(17)

# PRESENTATION AND COMPARISON OF AN EXACT STRUCTURAL ANALYSIS CODE WITH THE MIT DESIGN METHOD AND THE COUPLED WALL APPROXIMATE DEFLECTION ANALYSIS PROCEDURE

by

Gerald D. King

BS, Civil Engineering - 1995
BS, Psychology - 1995

University of Illinois at Urbana / Champaign

Submitted to the Department of Civil and Environment Engineering
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Civil and Environmental Engineering
at the

Massachusetts Institute of Technology

February 1998



Signature of Author ______________________________
Department of Civil and Environmental Engineering
September 10, 1997

Certified by ______________________________
Jerome J. Connor
Professor, Civil and Environmental Engineering

Accepted by ______________________________
Joseph M. Sussman
Chairman, Departmental Committee on Graduate Studies

# PRESENTATION AND COMPARISON OF AN EXACT STRUCTURAL ANALYSIS CODE WITH THE MIT DESIGN METHOD AND THE COUPLED WALL APPROXIMATE DEFLECTION ANALYSIS PROCEDURE

by

Gerald D. King

Submitted to the Department of Civil and Environmental Engineering
on September 10, 1997 in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Civil and Environmental Engineering

ABSTRACT

The first section written for this presentation is a Matlab computer program that automates the Beam - Column Method as presented by Chen and Lui. This code enables the user to analyze any structural system by calculating exact deflections while taking into account lateral and torsional considerations, support displacements, temperature changes, and all forms of external loads. This first section is used as a foundation for the entire presentation. The second part of this presentation is a comparison of two approximate deflection analysis procedures. Using the deflections calculated by the Beam - Column code found in Part I, the accuracy of the MIT Design Method presented by Connor and the accuracy of the Coupled Wall Method written by Stafford Smith, Kuster, and Hoenerkamp are evaluated and compared. This is accomplished by applying all three methods to an X - braced and a K - braced model frame with an aspect ratio of 7:1. Second order analyis considerations are briefly outlined and explained by way of discussing the underlying process behind each method.

Thesis Supervisor: Jerome J. Connor
Title: Professor of Civil and Environmental Engineering

## *Table of Contents*

*Chapter 1*

# INTRODUCTION

The process of estimating deflections in tall, steel-braced frames has often been a computationally demanding and lengthy process. In the following sections, two deflection approximation methods are introduced and outlined. These methods were designed to help the structural engineer in the initial steps of the design process when the external loads were known, and member size estimations were needed for the preliminary design of the system layout. Deflection estimates were also needed to ensure that the system behaved within certain serviceability constraints. Even if the resources used for obtaining exact deflection calculations could be spared, in the initial stage of the design process only estimates, not exact calculation of deflection, were needed.

The efficiency and accuracy of two approximate methods, the MIT Design Method and the Coupled Wall Method, are evaluated through a comparison of its results. Two model structures, whose physical characteristics are outlined in Appendix I, were constructed. After imposing a relatively light uniform wind load, both approximate methods were used to estimate the deflection of the structure and estimate its behavior. These results were then compared to the exact deflection as calculated by the Beam - Column code outlined in Appendix VII. The results given by the Matlab code were verified with the results that were found using SAP90, a finite element structural analysis code. By explaining the SAP90 analysis procedure, various issues regarding second order analysis techniques are outlined and discussed.

The organization of the paper is designed to introduce and familiarize the reader to the two approximate deflection methods presented in Chapters 2 and 3. The technical philosophy underlying the methods is explained, and the equations are introduced. A brief description of the SAP90 structural analysis computer package is given along with a basic discussion of second order nonlinear analysis procedures in Chapters 4 & 5. The Beam - Column Method is explained in detail in Chapter 6. Accuracy, quickness, ease of use, and demands on computational resources were all included in the criteria used for this comparison, and the results and conclusions are detailed in Chapters 8 & 9.

*Chapter 2*

# COUPLED WALL METHOD - STAFFORD SMITH, KUSTER, AND HOENDERKAMP

## Introduction

The first approximate method used in the comparison was outlined in a 1981 journal article entitled, "A Generalized Approach to the Deflection Analysis of Braced Frame, Rigid Frame, and Coupled Wall Structures." The authors, B. Stafford Smith, M. Kuster, and J. C. D. Hoenderkamp, presented a relatively uncomplicated method of estimating deflections in tall structures. Braced frames, rigid frames, and coupled walls were modeled as cantilevers whose deflection could be defined by their bending and shear characteristics.

The bending, or flexural component, of the structure's deflection is dependent on the combination of the overall composite flexure of the entire system and the bending of each individual member. The shear characteristics of a braced frame are dependent on the axial deformation of the diagonal members. Contraflexure of the columns and the beams also plays a minor role in the frame racking action. In a tall braced frame (structures with an aspect ratio of at least 1:6), the deflected shape is predominantly flexural. In shorter structures, the deflected shape is based on the shearing action. Coupled shear wall deflection is a combination of the two extremes; the behavior of a coupled wall takes into account a combination of both shearing and bending. This idea serves as the basis behind the approximate method that is detailed below.

## Equations

The deflection of a coupled shear wall is expressed as:

$$\boxed{y = \frac{wH^4}{EI_g}\left[\left\{\frac{1}{8} - \frac{1}{6}\left(\frac{x}{H}\right) + \frac{1}{24}\left(\frac{x}{H}\right)^4\right\} + \frac{1}{(k^2-1)}\left\{\frac{1-\left(x/H\right)^2}{2(k\alpha H)^2} + \frac{\cosh k\alpha(H-x) - 1 - k\alpha H(\sinh k\alpha H - \sinh k\alpha x)}{(k\alpha H)^4 \cosh k\alpha H}\right\}\right]}$$

The dimensionless parameters $\alpha$ and $k$ differentiate braced frames from coupled walls and rigid frames. The term $\alpha^2$ is the ratio of the racking shear rigidity of the overall structure and the flexural capabilities of the uncoupled vertical members. This variable can be expressed as $\alpha^2 = GA/EI$. The shear rigidity of the coupled wall is dependent on the bending rigidity of the connecting beams and the width and spacing of the walls. The GA term is equal to $Ph/\delta$ and is different for braced frames, rigid frames, and coupled walls.

For braced frames arranged in the X configuration, GA is equal to:

$$\boxed{GA = \frac{2hl^2E}{\left[\frac{h^3}{A_c} + \frac{(l^2 + h^2)^{3/2}}{A_d}\right]}}$$

and for frames arranged in the K configuration, GA is equal to:

$$\boxed{GA = \frac{2hl^2E}{\left[\frac{h^3}{A_c} + \frac{(l^2 + h^2)^{3/2}}{A_d}\right]}}$$

In this method, the relative height of this structure is measured in terms of the above variables. In structures with a large value of $\alpha H$ ($<100$) and $k^2$ approximately equal to one ($\sim 1$), the deflection of the structure will mirror a flexural curve. Braced frames and coupled walls with stiff connecting beams behave compositely and deflect in this shape. When the $\alpha H$ of a structure is near unity, the bending results in a forward flexural curve. This occurs when the bending of each individual vertical member (i.e. coupled wall systems with flexible beam connections) resists the lateral load. In structures where the $\alpha H$ lies between 2 and 80, the curve shape is dependent on the $k^2$ term. If the $k^2$ term is near unity, the structure will adopt a shear profile. If, on the other hand, the $k^2$ term is greater than or equal to 1.2, the structure will deflect flexurally. Structures with an intermediate degree of connectivity between the vertical members will show this type of behavior.

This Coupled Wall Method presents a process for determining deflection of rigid frames, coupled walls, and three types of braced frames (X braced, K braced, and single diagonal braced frames) subject to three different types of loading conditions (uniform, triangular, and point loads).

*Chapter 3*

# MIT DESIGN METHOD - CONNOR

## 1994 Edition

The second approximate method used in this comparison was originally presented in a paper published in November of 1994. *A New Method for the Design of Tall Buildings: The MIT design Method*, written by C. C. Pouangare and J. J. Connor presented another method for the determination of deflections in tall and super-tall buildings. Similar to the method presented in the Stafford Smith, Kuster, and Hoenderkamp paper, the structure is modeled as a fixed cantilever beam. In this paradigm, the stiffness properties of the building are transformed into equivalent beam properties. Strength criteria are checked after the designing for serviceability because tall structures are governed by flexural and stiffness considerations and not by strength. The method calculates the required shear and bending rigidities given the loads applied to the structure and the desired deflected shape of the system.

This process is based upon various assumptions and is built around two basic equations.

$$\boxed{\gamma(x) = \frac{V(x)}{GA_s(x)}} \qquad \beta(x) = \frac{M(x)}{EI(x)}x$$

The first equation is derived from the rotation and deformation of a beam. Constant curvature throughout the length of the beam is assumed. The second equation assumes that $\beta(x)$ is linear and that the interstory deflection is constant.

In this process, the distributions of $\gamma(x)$, $\beta(x)$, $GA_s(x)$, and $EI(x)$ were calculated. Two assumptions were made. In the first case, the linear shear and bending rigidities were assumed to vary linearly; in the second case, the shear rigidity distribution was assumed to be constant. Problems were found in both cases.

Assuming a linearly varying bending rigidity ($GA_s$) that decreased inversely with height would mean that the shear deformation, $\gamma(x)$, would increase linearly. This would mean that the rotation, $\beta(x)$, would also decrease linearly. Since the slope of the rotation must be positive, a linearly decreasing rotation was physically impossible. As a result, it was evident that $\gamma(x)$ should always decrease with x.

The second assumption had inherent problems as well. In order to obtain a constant shear rigidity distribution, shear deformation would have to decrease as a function of x. This meant that at the origin the slope of the rotation would be equal to zero and the stiffness of the beam would be infinite. This is another physical impossibility.

As a result of these problems, the distributions of $\gamma(x)$, $\beta(x)$, $GA_s(x)$, and $EI(x)$ were modified and presented as an assumed distribution. The process of calculating deflections was then completed based partially on these assumed characteristics. The resulting method was adequate for estimating deflections in tall braced frames.

## 1996 Revised Edition

In 1996, Prof. J. J. Connor and his student, Boutros Klink, revised the MIT Design Method and presented it in their book, *Introduction to Motion Based Design*. This new method avoided the assumptions and the resulting problems used in the original process. The new set of equations proved to be far more elegant and their derivations were much simpler and straightforward.

The two equations involving bending and shear rigidities again were the foundations for the modified procedure. The method was based on the projection of a desired deflection curve. The strains found in the chords and diagonals were related to the deformations through a series of constitutive relations. Finally, by equating the shear and bending moment distributions to the related rigidity distributions, the areas of all key members can be estimated. Specifications for the beams, columns, and diagonals can then be derived from the imposed load and the desired deflected shape.

# Equations

The MIT Design Method can be used to estimate deflection in tall steel frames braced in both the X and the K configurations. For each configuration, the two equations for the rigidity distributions remain the same. They are as follows:

$$\boxed{D_B = \frac{bH^3}{4s\gamma^*}\left[1-\frac{x}{H}\right]^2}$$

$$\boxed{D_T = \frac{bH}{\gamma^*}\left[1-\frac{x}{H}\right]}$$

For the X configuration, the strain distribution is as follows:

$$\boxed{D_B = \frac{A^C E^C B^2}{2}}$$

$$\boxed{D_T = A^D E^D \sin(2\theta)\cos(\theta)}$$

As for the K configuration, the distribution follows:

$$\boxed{D_B = \frac{A^C E^C B^2}{2}}$$

$$\boxed{D_T = \frac{1}{\frac{2L^3}{lB^2 A^D E^D} + \frac{l^2}{2B^2 A^C E^C} + \frac{B}{4lA^B E^B}}}$$

*Chapter 4*

# SAP90 - STRUCTURAL ANALYSIS PROGRAM

## Introduction

SAP90 is the latest edition of a series of structural analysis programs written by Professor Edward Wilson at the University of California, Berkeley. The development of the series has taken place over a span of more than 25 years resulting in a very powerful finite element analysis program. It is the most reputable and widely used computer program in the field of structural analysis.

The element models used by SAP90 can take four different forms:

1. Frame elements, which are used to represent
   - two - and three - dimensional frame systems
   - two - and three - dimensional truss systems
2. Shell elements, used for
   - three dimensional shell structures
   - two - and three - dimensional membrane systems
   - two - and three - dimensional plate bending systems
3. Solid elements, use for
   - three - dimensional solid structures, and
4. ASolid elements, which are used for
   - three - dimensional plane - strain structures
   - two - dimensional plane - stress structures
   - three - dimensional axisymmetric structures.

## Static Analysis

The static analysis performed by SAP90 involves solving the set of linear equations represented by:

$$\mathbf{K\,U = R}$$

where **K** is the stiffness matrix,
**U** is the displacement vector, and
**R** is the vector of applied nodal loads and fixed end forces.

The frame elements can be subjected to loads in the form of

- Gravity loading
- Span uniform loading
- Span point loads
- Span trapezoidal loading
- Thermal loading, and
- Prestress loading

## Deflected Shape

SAP90 assumes that the deflected shape of each member is cubic for the bending and linear for shear. The actual deflected shape may vary from this generalization in two different situations:

- The member is non-prismatic. SAP90 handles this situation by averaging the properties of both ends of the member. The result is a good, but not exact, approximation.

- The existence of loads acting on the member. This situation could arise due to temperature changes, prestressing, or the inclusion of self-weight. In this case, the program computes the fixed end forces applied to each end of the member. The deflected shape is calculated using these loads.

The exact shape of the deflection is described by stability stiffness equations that are trigonometric for compression forces and hyperbolic for large tension forces. These functions ($\phi$ factors) are also used in the Beam - Column Method, and the different factors used for compression loads and tension loads are presented below:

**Tensile Axial Force**

$$\Phi_1 = \frac{(kL)^3 \sinh(kL)}{12\Phi_t}$$

$$\Phi_2 = \frac{(kL)^2[\cosh(kL) - 1]}{6\Phi_t}$$

$$\Phi_3 = \frac{(kL)[kL\cosh(kL) - \sinh(kL)]}{4\Phi_t}$$

$$\Phi_4 = \frac{(kL)[\sinh(kL) - kL]}{2\Phi_t}$$

$$\Phi_t = 2 - 2\cosh(kL) + (kL)\sinh(kL)$$

where $k = \sqrt{\frac{P}{EI}}$

**Compressive Axial Force**

$$\Phi_1 = \frac{(kL)^3 \sin(kL)}{12\Phi_c}$$

$$\Phi_2 = \frac{(kL)^2[1 - \cos(kL)]}{6\Phi_c}$$

$$\Phi_3 = \frac{(kL)[\sin(kL) - kL\cos(kL)]}{4\Phi_c}$$

$$\Phi_4 = \frac{(kL)[kL - \sin(kL)]}{2\Phi_c}$$

$$\Phi_c = 2 - 2\cos(kL) - (kL)\sin(kL)$$

where $k = \sqrt{\frac{P}{EI}}$

## Nonlinear Behavior

SAP90 is also capable of taking into consideration the effects of an axial load on the transverse bending behavior of the frame elements. It is very important to take this P- delta effect, a type of geometric nonlinearity, into account during the analysis of gravity loads on the lateral stiffness of tall structures.

In the initial stages of loading, the structure behaves linearly; the system's load-deflection relationship is linear. Assuming an initial linear behavior, SAP90 forms the basic equilibrium equations using the undeformed geometry of the structure. These linear equations are independent of both the load imposed on the system and the resulting deflection of the structure. This allows the user to superimpose different loads during the computation of the deflections resulting in a high degree of computational efficiency and a reduction of the demands placed on the equation solving system.

If the load-deflection relationships become skewed, the structure will exhibit nonlinear behavior, which can be caused by three different factors:

1. Geometric or kinematic nonlinearity (Large - stress effect)
2. Geometric or kinematic nonlinearity (Large - displacement effect)
3. Material nonlinearity

## Geometric Nonlinearity

**Large-stress effect**: when large stresses (i.e. forces and moments) are imposed on a member or structure, the equilibrium equations for the deformed and undeformed geometries might be significantly different regardless of the resulting deformations. P-Delta effects are an example of this type of nonlinearity.

**Large-displacement effect**: when the system undergoes large deformations (i.e. large strains and rotations), the equilibrium equations must be redefined for the deformed geometries because the simple stress and strain principles are no longer applicable.

The geometric nonlinearity associated with our model structure stems from a significant moment that originates from the application of a large direct force upon a small deflection parallel to the undeformed direction of the member. This nonlinearity ultimately affects the behavior of the member or structure. If the deflection acted upon is small, then the resulting moment is proportional to the magnitude of the deflection. The forces that typically create the P-Delta effects usually act in tension or compression, but not in shear.

Geometric nonlinearity is especially important for tall, slender structures and members subjected to large gravity loads. Analysis of geometric nonlinearity can be carried out by using the stability stiffness functions in the Beam-Column Method or by using an initial stress stiffness matrix in a finite element analysis. Each method is exact and should provide the same results.

## Material Nonlinearity

When a material is strained past its proportional limit, before it reaches its ultimate load carrying capacity, the stress-strain relationship is no longer linear. Material nonlinearity can affect the behavior of a structure even when the equilibrium equations for the original geometry still holds true. For steel braced frames, material nonlinearity occurs when the cross section yields along the member length. This takes place as the initial yield moment ($M_y$) increases to the full plastic moment ($M_p$).

There are two models that take into account the effects of material nonlinearity:

1. Concentrated plasticity (plastic hinge) model - ignores the progressive yielding that occurs in the cross-section and along the member.

2. Distributed plasticity (plastic zone) model - more complex model that considers the spread of the yielding in the cross-section and along the member.

Material nonlinearity is not taken into account by any of the deflection approximation methods compared in this paper or in the SAP90 deflection computations.

## Method

In the process of solving for P-Delta effects, SAP90 constructs a stiffness matrix for the entire system, imposes all the forces, and analyzes the structure iteratively. This process is generalized as follows:

1. Compute the initial elastic stiffness matrix (equilibrium equations)
2. Equate vector forces to zero
3. Apply loads
4. Calculate resulting displacements and axial forces
5. Modify stiffness matrix to take into account P-Delta effects
6. Repeat steps 3 - 5 until displacements converge.

Two constraints specified by the user, relative displacement tolerance and the maximum number of iterations, control this "direct iteration" procedure.

The **relative displacement tolerance** often constrains the process if the initial deflection estimate is relatively accurate. If the difference between the displacements is lower than the value specified the iteration process is terminated. This change in displacement, which includes both rotational and translational movements, is defined as a ratio of the maximum difference in the displacements to the largest displacement in each iteration.

The **maximum number of iterations** can be used to stop the program if the results of each iteration fail to converge. This serves to limit computation time and allows the user to reset the program with more appropriate specifications.

Failure to converge can stem from various causes:

- Number of iterations is too small to provide meaningful results. 2 - 5 iterations are reasonable. More iterations might be used depending on the complexity of the problem.

- Convergence tolerance is either too small or too large. A tolerance that is too small will converge slowly so that the analysis will take a long time to complete. An unreasonably large tolerance will result in few iterations being performed yielding meaningless results.

- The load imposed on the structure is near critical. At this point the members are near buckling and the loads acting upon the structure should be decreased.

The method used by the SAP90 program is very similar to the process outlined by the Newton - Rhapson method. Variations of this method are briefly summarized in the following section.

*Chapter 5*

# SOLUTION ALGORITHMS FOR NONLINEAR ANALYSIS

## Basic Concepts

In a first order analysis, the equilibrium and kinematic relationships are based on the original, undeformed geometry of the system. In a second order analysis, these equations are based on the deformed geometry of the system. A second order analysis is needed to determine the stability aspects of the structure.

A first order analysis is a relatively simple process, and results can be gathered immediately from one iteration. A second order analysis is much more complicated requiring iteration upon iteration to gather meaningful results. These iterations are needed because the deformed geometry of the structure is unknown during the initial formulation of the equilibrium relationships. During one iteration, the load is imposed on the member or structure, and then the displacements and deformations are calculated. New relationships are established in accordance to this new geometry, the process is repeated again with the re-application of loads to the structure. Iterations can follow one of four different schemes as listed below, but only the Load Control Method will be described in more detail.

1. Load Control Method
2. Displacement Control Method
3. Arc length Control Method
4. Work Control Method.

The **Load Control Method**, one of the oldest methods used for nonlinear analysis, calculate the amount of load imposed on the structure as a fraction of the total applied load. This incremental loading often produces a drift - off error equal to the difference between the external applied forces and the internal forces of the structure. The source of this discrepancy is the linearization process that uses a stiffness matrix based on the original configuration of the structure. The Newton - Rhapson method is used in combination with the load control method to eliminate the unbalanced forces between each by reiterating the stiffness matrix at each load increment imposed on the system.

The Newton - Rhapson Load Control Method serves as the foundation for the nonlinear solver in both the SAP90 structural analysis program and the Beam - Column Matlab code. Variations in the implementation of this method for both programs have been and will be described again in each corresponding section of this paper.

*Chapter 6*

# BEAM - COLUMN METHOD

## Introduction

The deflections found using both of the aforementioned processes were checked with a 2 - dimensional structural analysis paradigm. The Beam - Column Method was chosen because of its ability to take into account geometrical nonlinearity and because it yields exact results. The Beam-Column Method, which is taught in most structural analysis classes, was automated by codifying the procedures. The program is listed in Appendix VII.

## Method

Matlab was chosen as the language for this program because of its power to manipulate matrices. The organization of the code follows the procedure as outlined in various structural analysis classes. The first step in this process is to collect data from each node and member. Such information includes:

- the location of each node
- the axial, shear, moment, torsion, and bi-moment forces
- type of restraints on all five degrees of freedom,
- the displacement in the five degrees of freedom
- spring forces in each degree of freedom, if applicable
- positive and negative nodes assignments for each member
- the physical properties (Area, E, I $I_w$, and $k_v$) of each member
- the in-span load information.

The length and inclination angle of each member are then computed, and the direction matrix of each member is calculated and stored. In the development of the stiffness matrix, the axial force must be calculated to see if it is compressive or tensile. In order to collect this information, a stiffness matrix for each member is needed for the computation of forces due to displacement effects. In the initial case, no axial forces were assumed and as a result, sii = 4 and sij = 2. The forces are found, and both an elemental and global load vector is formed for the entire structure. Variables used to denote the various forces are as follows:

- FMSIX_NF - Forces on Frame Nodes
- FMSIX_DISP - Forces due to Support Displacement
- FM_TEMP - Forces due to Temperature Changes
- FMSIX_FEF - Fixed End Forces

After formulating the force vector, axial loads are determined to be compressive or tensile which is important in deciding which stability stiffness function ($\phi$ factor) to use. These equations are the same phi equations presented in Chapter 4.

The basic concept behind this method is founded on the relationship between forces and the end displacement and is derived from the slope deflection equations. Appendix II shows the derivation of these equations for a system with three degrees of freedom per node.

The final stiffness matrix is as follow:

$$k = T^T \tilde{k} T = \frac{EI}{L}\begin{bmatrix} \frac{A}{I} & 0 & 0 & -\frac{A}{I} & 0 & 0 \\ 0 & \frac{12}{L^2}\phi_1 & \frac{6}{L}\phi_2 & 0 & -\frac{12}{L^2}\phi_1 & \frac{6}{L}\phi_2 \\ 0 & \frac{6}{L}\phi_2 & 4\phi_3 & 0 & -\frac{6}{L}\phi_2 & 2\phi_4 \\ -\frac{A}{I} & 0 & 0 & \frac{A}{I} & 0 & 0 \\ 0 & -\frac{12}{L^2}\phi_1 & -\frac{6}{L}\phi_2 & 0 & \frac{12}{L^2}\phi_1 & -\frac{6}{L}\phi_2 \\ 0 & \frac{6}{L}\phi_2 & 2\phi_4 & 0 & -\frac{6}{L}\phi_2 & 4\phi_3 \end{bmatrix}$$

After completing the formulation of the 6 x 6 stiffness matrix, warping and torsion are by forming a 4 x 4 matrix. Appendix III outlines the process for determining this matrix. When completed, the final matrix takes into account five degrees of freedom per node. It follows that a 10 x 10 matrix represents the stiffness and stability of each member.

## Two Cycle Iterative Method

The memory capacity of the machine used to run such a computationally demanding code for a large structure subject must be extremely large. To illustrate this point the 7 story, K braced model structure used in this presentation has 56 nodes. The resulting global stiffness matrix would have over 78,400 terms. Because of the huge memory capacity needed to compute a frame with many members, a full scale Newton - Rhapson method can not be carried out. Instead, a two cycle iterative method is used.

The first cycle, a first order analysis, is performed on the system when the code determines the axial loads on each member. The K stiffness matrix is obtained without considering any second order effects. After the axial forces are computed, the program carries out a second order analysis with the equations listed above. The global stiffness matrix is then updated to take into account the second order effects that arise from geometrical nonlinearity. Unlike the Newton - Rhapson method where incremental loads are imposed on the system, the full load is used in both cycles.

Al-Mashary and Chen (1990a) have proved the validity of this two cycle iterative method. By testing five different frames and analyzing the results, it was shown that the methods and the resulting differences in deflections were well within the acceptable limits when compared to more rigorous structural analysis methods that include commercially available structural analysis tools such as SAP90.

## *Chapter 7*

# COMPARISON

## Process

The process used to compare the three deflection analysis methods is outlined below:

1. Two structural models developed - aspect ratio 7 to 1
   - One model braced in the X configuration
   - One model braced in the K configuration
2. Member size distributions calculated using the modified MIT Design Method equations
3. Beams, columns, and bracing sizes adjusted to the largest sizes recommended in step 2
4. Loading scheme developed and imposed on the structure
5. Structure analyzed using the MIT Design Method
6. Structure analyzed with the Coupled Wall Method
7. Structure analyzed with the Matlab Beam-Column Code
8. Results in step 7 confirmed using SAP90
9. Absolute and inter-story deflections calculated with each process were plotted and compared
10. Member sizes adjusted and re-analyzed
11. Plots developed to show effects of changing column and bracing sizes.

## Member Size Selection

In structural engineering, the efficiency of the design process has always been burdened by the need to make initial guesses from which the final solutions are derived through a series of iterations. The length of this process is dependent on this first step; an accurate preliminary guess requires few, if any, design iterations. The structural design objectives always include such criteria such as serviceability and strength, and the task of the engineer is to design a structural system that behaves within the boundaries of these criteria. By selecting the member sizes and the materials used, the engineer is able to mold the physical characteristics of the system and control the resulting behavior of the structure.

Traditionally, the design of a structural system begins with a number of guesses. Often times, member sizes and the materials used are somewhat constrained by architectural requirements and the engineers are required to pick member properties that fit this criterion. This constraint gives the engineer a range of member sizes he or she can use; however, an exact process that calculates accurate member sizes has never been developed. Given a deflection objective and the characteristics of the material used, the engineer is required to make an educated guess on the size of the members used in the structural system. The engineer's design experience and intuition influences the accuracy of this initial guess. The length of the remaining design process depends on how close these preliminary sizes are to the optimal member sizes. Deflections are calculated based on these initial member selections, and if the structural deflections exceed the serviceability criteria, an iteration is performed to adjust the member sizes. These iterations are performed until the deflections of the system meet the serviceability criteria.

The MIT Design Method allows the engineer to bypass the time-consuming member sizing iteration process. After entering the material characteristics and deflection constraints, exact member sizes can be calculated for the beams, the columns, and the braces with two sets of equations. Depending on the frame configurations, the equations are as follows:

for the X - Braced configuration

$$\boxed{A^D = \frac{bH}{\gamma * E^D \sin(2 \cdot theta)\cos(theta)}\left[1 - \frac{x}{H}\right]}$$

$$\boxed{A^C = \frac{2bH^3}{4s\gamma * E^C B^2}\left[1 - \frac{x}{H}\right]^2}$$

and for the K - Braced configuration

$$\boxed{A^D = \frac{2L^3}{lB^2E^D\left(\left(\frac{1}{\frac{bH}{\gamma *}}\left[1 - \frac{x}{H}\right]\right) - \left(\frac{l^2}{2B^2A^CE^C}\right) - \left(\frac{B(Scale)}{4E^BlA^C}\right)\right)}}$$

$$\boxed{A^C = \frac{2bH^3}{4s\gamma * E^C B^2}\left[1 - \frac{x}{H}\right]^2}$$

*Chapter 8*

# RESULTS

## List of Figures



In this section, the calculations of both methods are compared to the actual deflection behavior of both the X - braced and the K - braced systems. The magnitude of the differences are presented and the deviations and deflection differences (in/ft) are shown graphically.



The actual and the estimated behavior of the structure are shown graphically. The center graph on each page shows the behavior of the frame with model member sizes. The graphs on either side of the page show the behavior of the structure when column or beam sizes are modified. These are included to show how each estimation is affected by changing different member sizes.

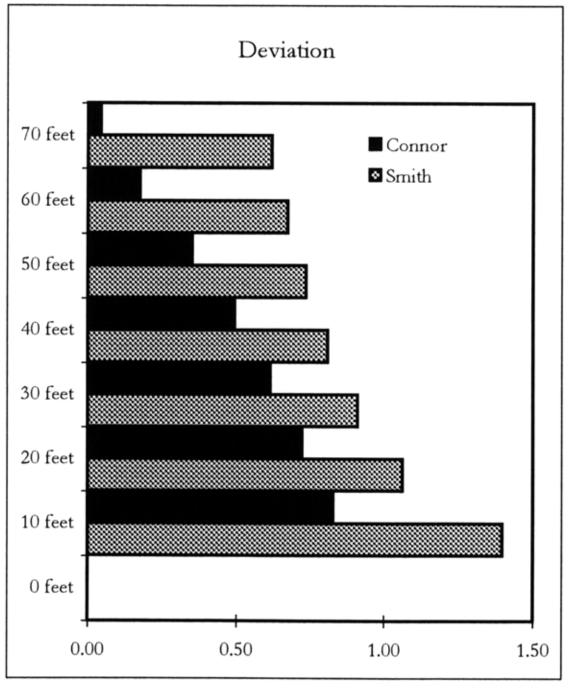


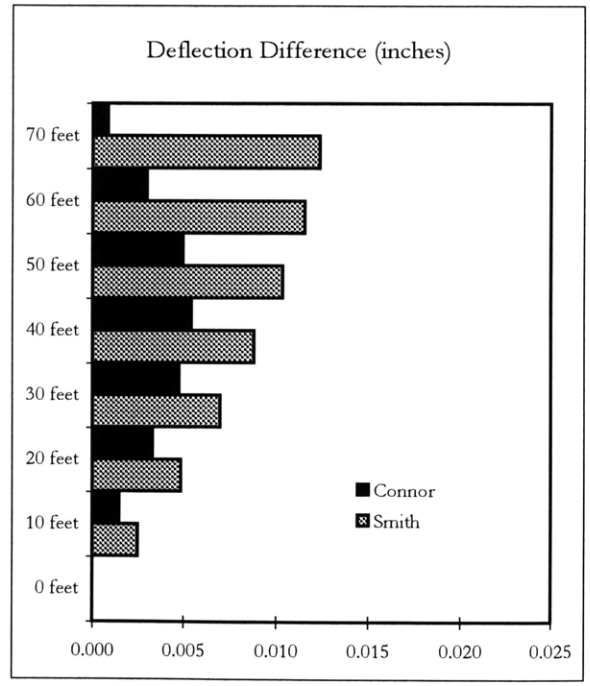
In this section, the results from changing the column and beam sizes are illustrated in a way so that one can compare the deflections of each change to all other changes applied to the structure. Each bar graph contains five columns. Four of the columns represent changes to the beams and the columns. The middle column represents the deflection of the model structure which is standardized to unity.

## Stafford Smith Connor Comparison - X Bracing

| Height | Deflection (inches) | | | Deflection (inches) / Height (feet) | | | Deviation | | Deflection Difference (in/ft) | |
|---|---|---|---|---|---|---|---|---|---|---|
| | SAP | Smith | Connor | SAP | Smith | Connor | Smith | Connor | Smith | Connor |
| 0 feet | 0.00000 | 0.00000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 10 feet | 0.12445 | 0.29823 | 0.02160 | 0.001778 | 0.004260 | 0.000309 | 1.396375 | 0.826450 | 0.002483 | 0.001469 |
| 20 feet | 0.31912 | 0.65817 | 0.08844 | 0.004559 | 0.009402 | 0.001263 | 1.062494 | 0.722874 | 0.004844 | 0.003295 |
| 30 feet | 0.53744 | 1.02564 | 0.20663 | 0.007678 | 0.014652 | 0.002952 | 0.908361 | 0.615527 | 0.006974 | 0.004726 |
| 40 feet | 0.76418 | 1.38227 | 0.38639 | 0.010917 | 0.019747 | 0.005520 | 0.808818 | 0.494370 | 0.008830 | 0.005397 |
| 50 feet | 0.98810 | 1.71380 | 0.64201 | 0.014116 | 0.024483 | 0.009172 | 0.734435 | 0.350264 | 0.010367 | 0.004944 |
| 60 feet | 1.20114 | 2.01004 | 0.99184 | 0.017159 | 0.028715 | 0.014169 | 0.673442 | 0.174254 | 0.011556 | 0.002990 |
| 70 feet | 1.39958 | 2.26533 | 1.45833 | 0.019994 | 0.032362 | 0.020833 | 0.618577 | 0.041976 | 0.012368 | 0.000839 |

| Height | Deflection (feet) | | |
|---|---|---|---|
| | SAP | Smith | Connor |
| 0 feet | 0.00000 | 0.00000 | 0.00000 |
| 10 feet | 0.01037 | 0.02485 | 0.00180 |
| 20 feet | 0.02659 | 0.05485 | 0.00737 |
| 30 feet | 0.04479 | 0.08547 | 0.01722 |
| 40 feet | 0.06368 | 0.11519 | 0.03220 |
| 50 feet | 0.08234 | 0.14282 | 0.05350 |
| 60 feet | 0.10010 | 0.16750 | 0.08265 |
| 70 feet | 0.11663 | 0.18878 | 0.12153 |

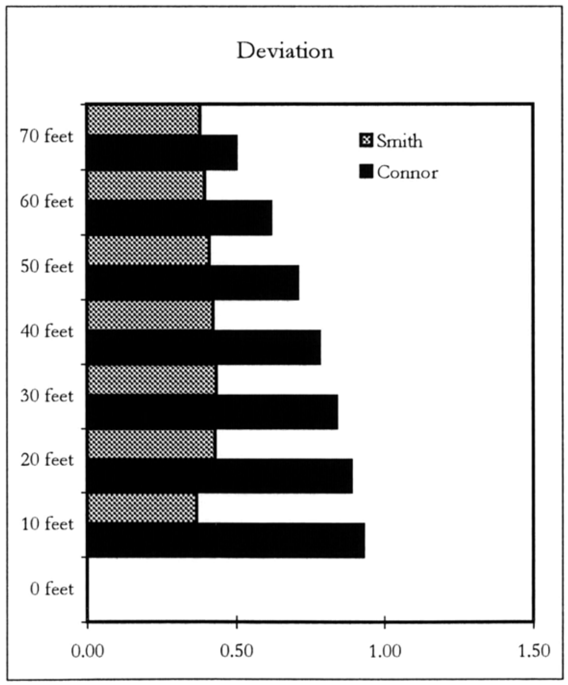


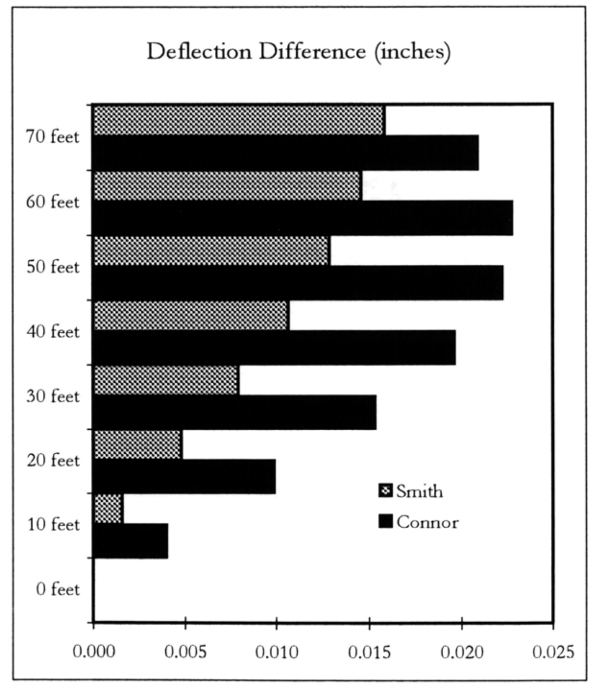



# Stafford Smith Connor Comparison - K Bracing

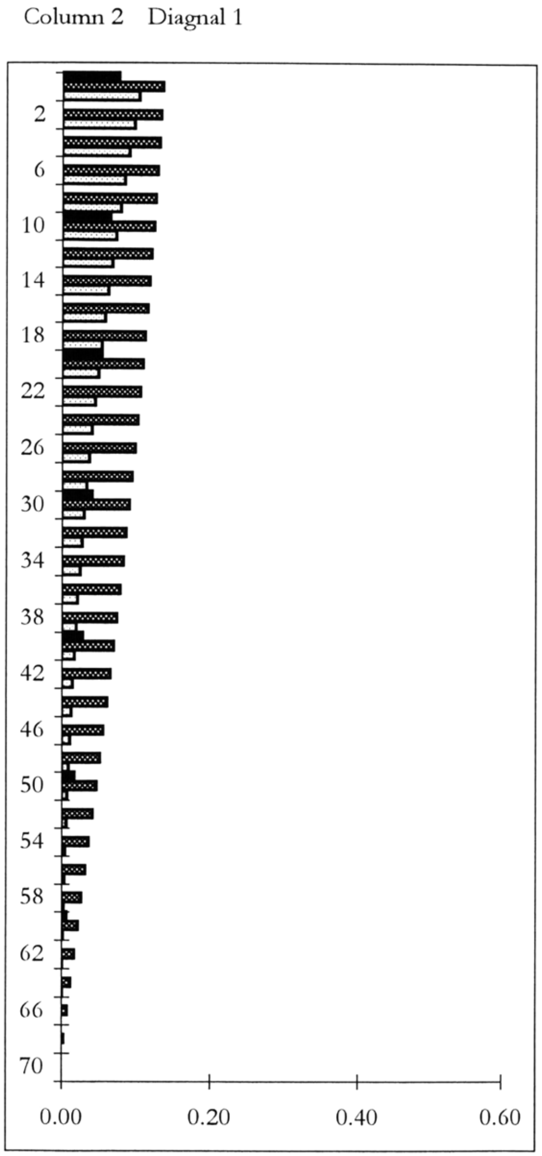
| Height | Deflection (inches) | | | Deflection (inches) / Height (feet) | | | Deviation | | Deflection Difference (in/ft) | |
|---|---|---|---|---|---|---|---|---|---|---|
| | SAP | Smith | Connor | SAP | Smith | Connor | Smith | Connor | Smith | Connor |
| 0 feet | 0.00000 | 0.00000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 10 feet | 0.30221 | 0.19105 | 0.02160 | 0.004317 | 0.002729 | 0.000309 | 0.367825 | 0.928531 | 0.001588 | 0.004009 |
| 20 feet | 0.77981 | 0.44464 | 0.08844 | 0.011140 | 0.006352 | 0.001263 | 0.429815 | 0.886593 | 0.004788 | 0.009877 |
| 30 feet | 1.27927 | 0.72508 | 0.20663 | 0.018275 | 0.010358 | 0.002952 | 0.433210 | 0.838476 | 0.007917 | 0.015323 |
| 40 feet | 1.75934 | 1.01403 | 0.38639 | 0.025133 | 0.014486 | 0.005520 | 0.423632 | 0.780376 | 0.010647 | 0.019614 |
| 50 feet | 2.19906 | 1.29722 | 0.64201 | 0.031415 | 0.018532 | 0.009172 | 0.410104 | 0.708054 | 0.012883 | 0.022244 |
| 60 feet | 2.58444 | 1.56445 | 0.99184 | 0.036921 | 0.022349 | 0.014169 | 0.394666 | 0.616228 | 0.014571 | 0.022751 |
| 70 feet | 2.91858 | 1.80975 | 1.45833 | 0.041694 | 0.025854 | 0.020833 | 0.379921 | 0.500328 | 0.015840 | 0.020861 |

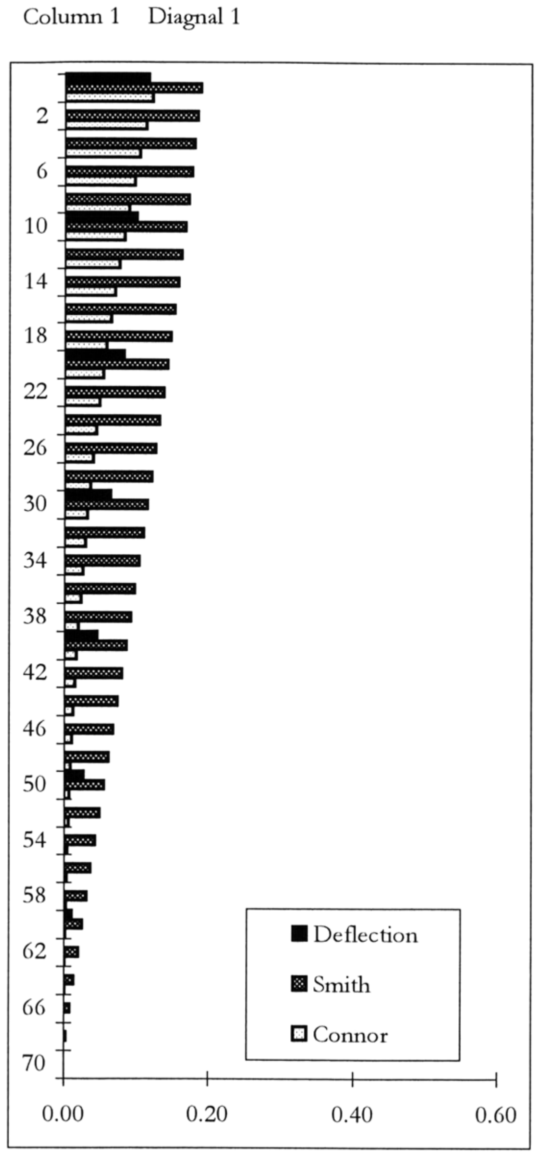
| Height | Deflection (feet) | | |
|---|---|---|---|
| | SAP | Smith | Connor |
| 0 feet | 0.00000 | 0.00000 | 0.00000 |
| 10 feet | 0.02518 | 0.01592 | 0.00180 |
| 20 feet | 0.06498 | 0.03705 | 0.00737 |
| 30 feet | 0.10661 | 0.06042 | 0.01722 |
| 40 feet | 0.14661 | 0.08450 | 0.03220 |
| 50 feet | 0.18326 | 0.10810 | 0.05350 |
| 60 feet | 0.21537 | 0.13037 | 0.08265 |
| 70 feet | 0.24322 | 0.15081 | 0.12153 |

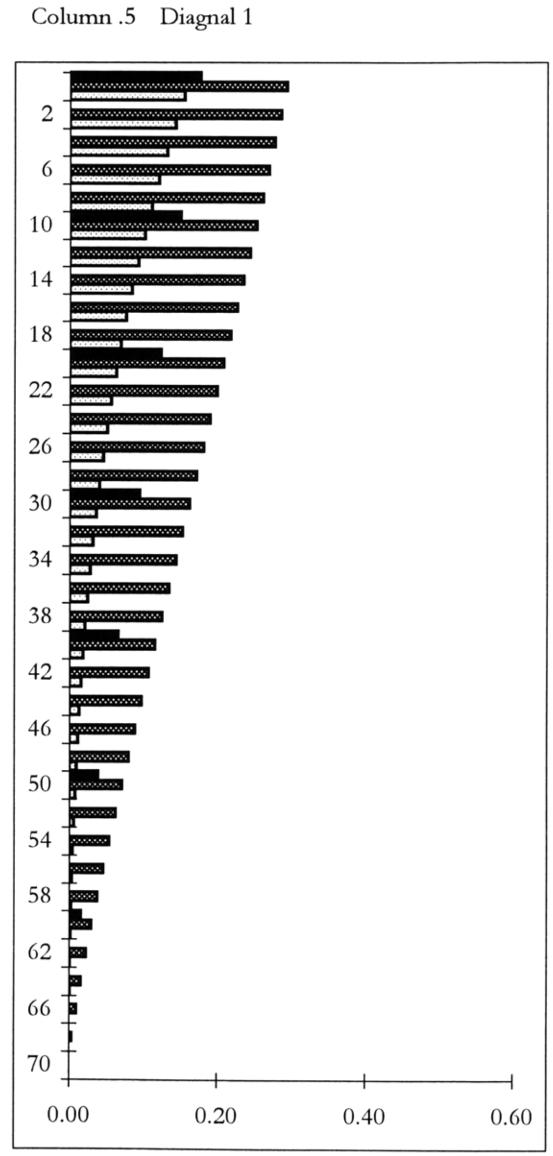


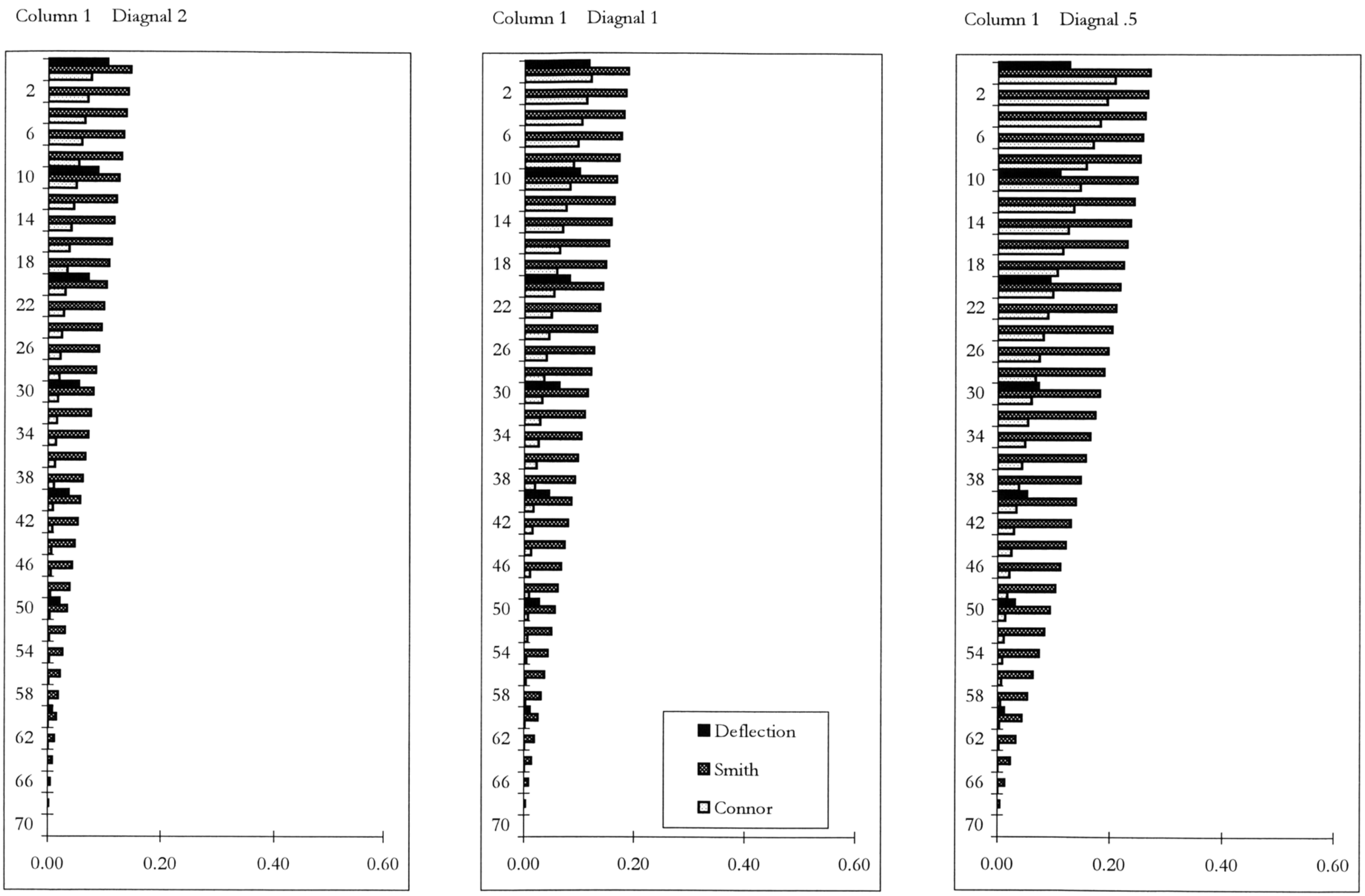
## Varying Column Sizes / X Braced



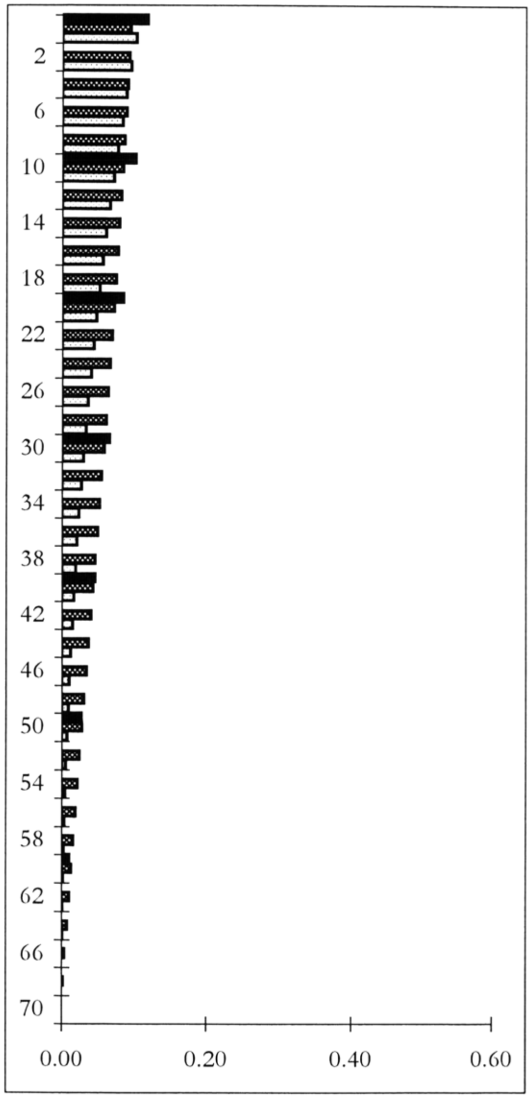



## Varying Diagonal Sizes / X Braced

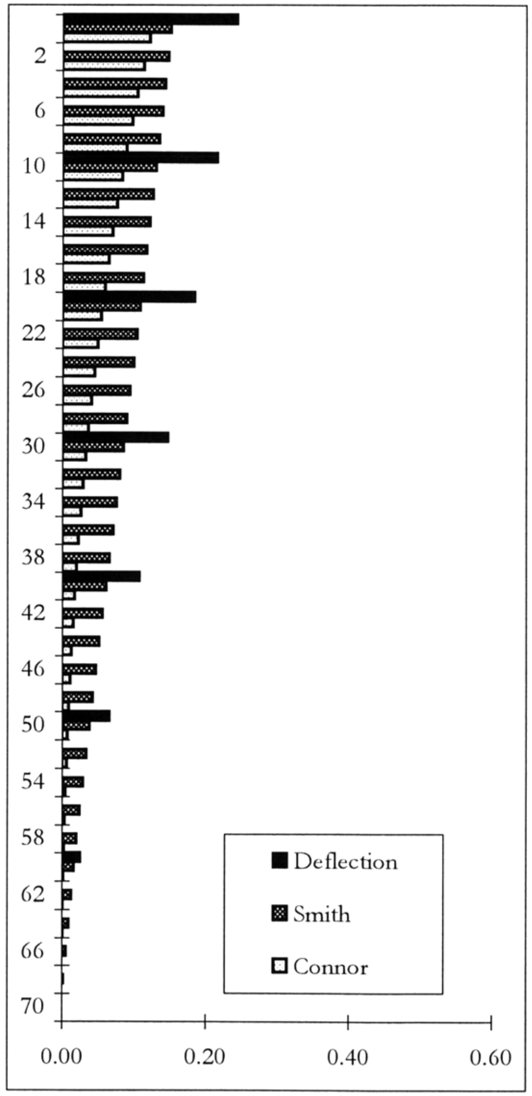


## Varying Column Sizes / K Braced

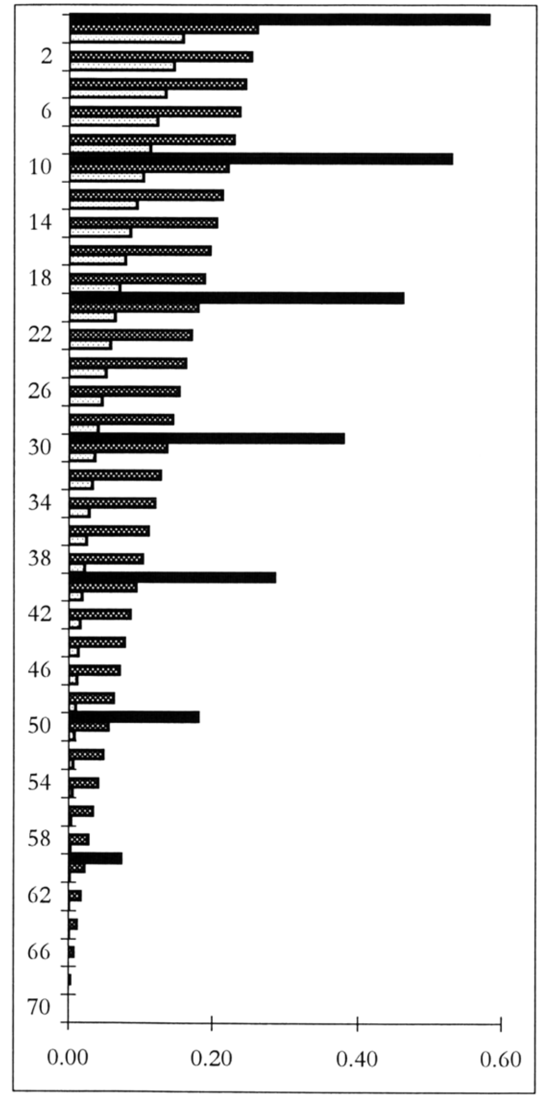
Column 2 Diagnal 1



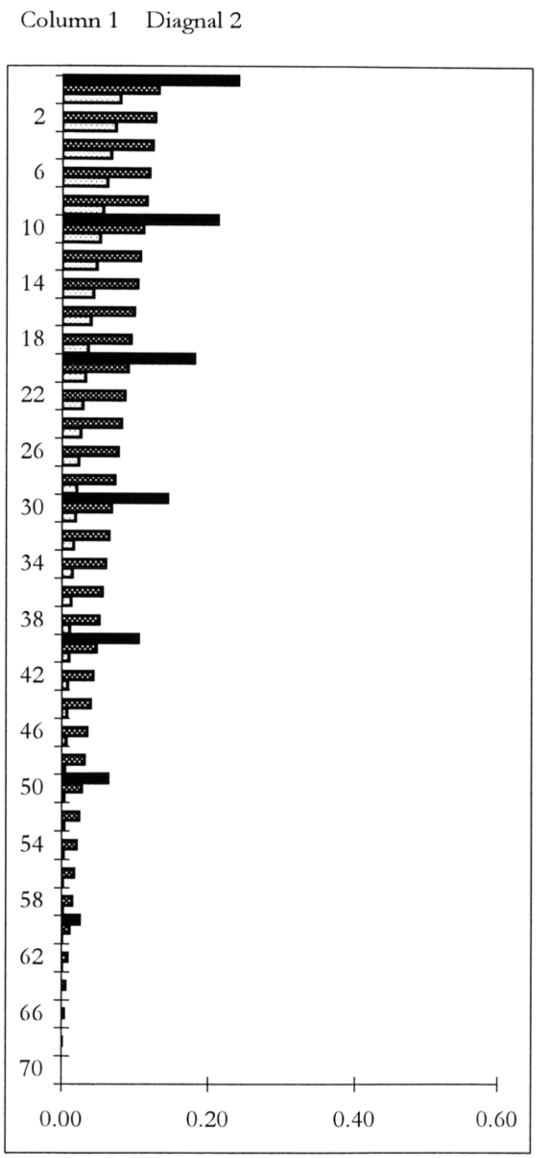

Column 1 Diagnal 1

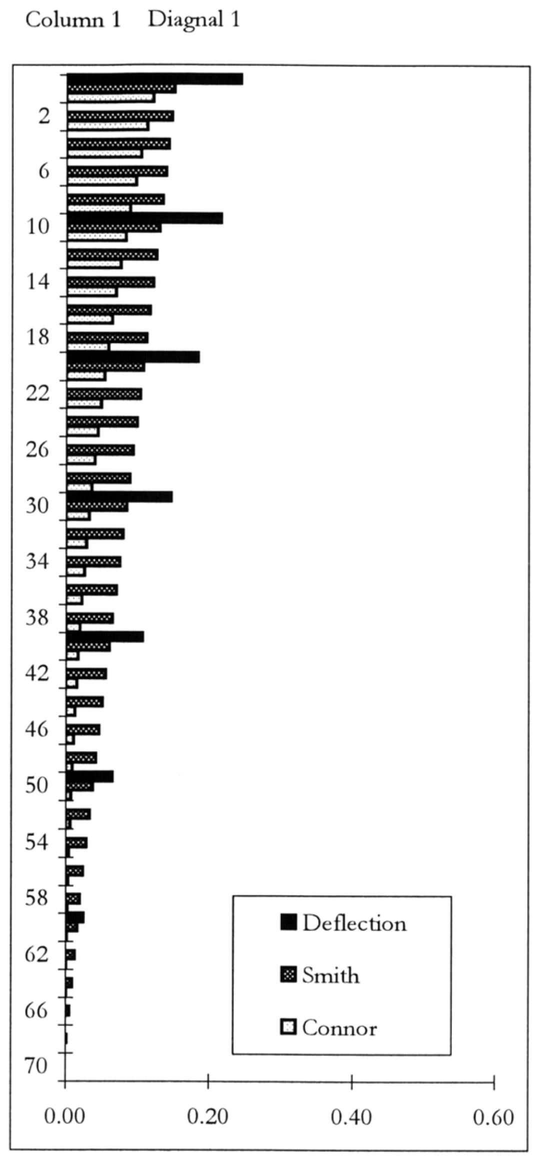


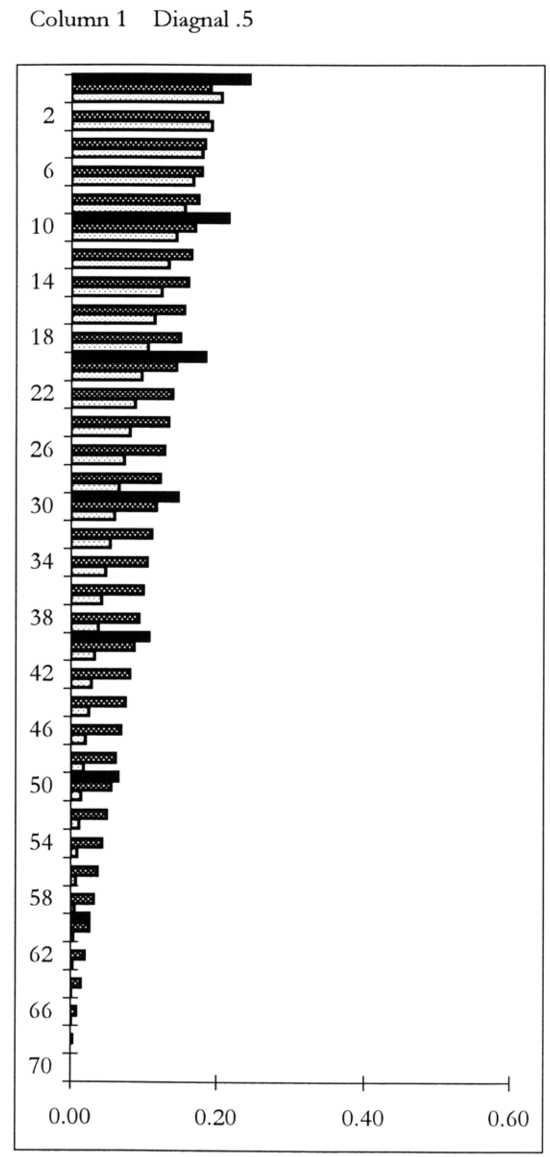

Column .5 Diagnal 1



## Varying Diagonal Sizes / K Braced



## Effects of Changing Diagonal and Column Sizes / X - Bracing

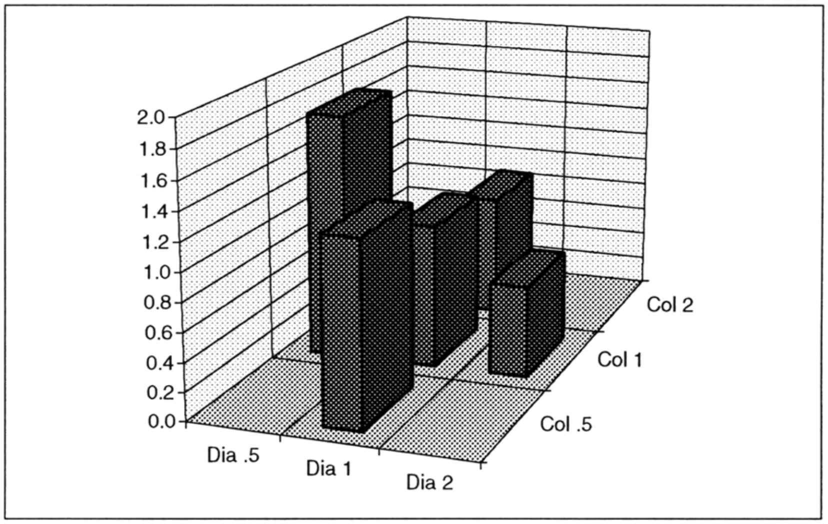
### MIT Design Method

Ratio

| | Dia .5 | Dia 1 | Dia 2 |
|---|---|---|---|
| Col .5 | | 1.280 | |
| Col 1 | 1.720 | 1.000 | 0.640 |
| Col 2 | | 0.860 | |

Deflection (ft)

| | Dia .5 | Dia 1 | Dia 2 |
|---|---|---|---|
| Col .5 | | 0.156 | |
| Col 1 | 0.209 | 0.122 | 0.078 |
| Col 2 | | 0.105 | |



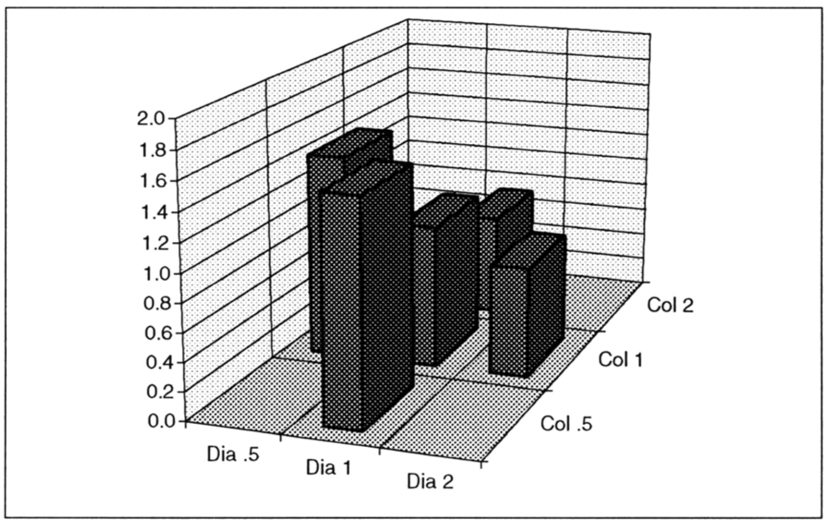

### Coupled Wall Method

Ratio

| | Dia .5 | Dia 1 | Dia 2 |
|---|---|---|---|
| Col .5 | | 1.555 | |
| Col 1 | 1.438 | 1.000 | 0.777 |
| Col 2 | | 0.719 | |

Deflection (ft)

| | Dia .5 | Dia 1 | Dia 2 |
|---|---|---|---|
| Col .5 | | 0.294 | |
| Col 1 | 0.272 | 0.189 | 0.147 |
| Col 2 | | 0.136 | |



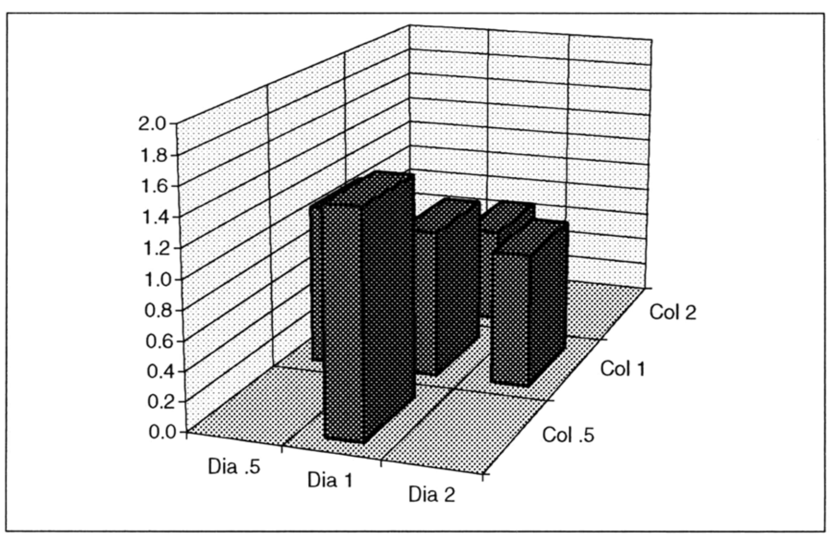

### Matlab Code

Ratio

| | Dia .5 | Dia 1 | Dia 2 |
|---|---|---|---|
| Col .5 | | 1.515 | |
| Col 1 | 1.100 | 1.000 | 0.908 |
| Col 2 | | 0.657 | |

Deflection (ft)

| | Dia .5 | Dia 1 | Dia 2 |
|---|---|---|---|
| Col .5 | | 0.177 | |
| Col 1 | 0.128 | 0.117 | 0.106 |
| Col 2 | | 0.077 | |



## Effects of Changing Diagonal and Column Sizes / K - Bracing

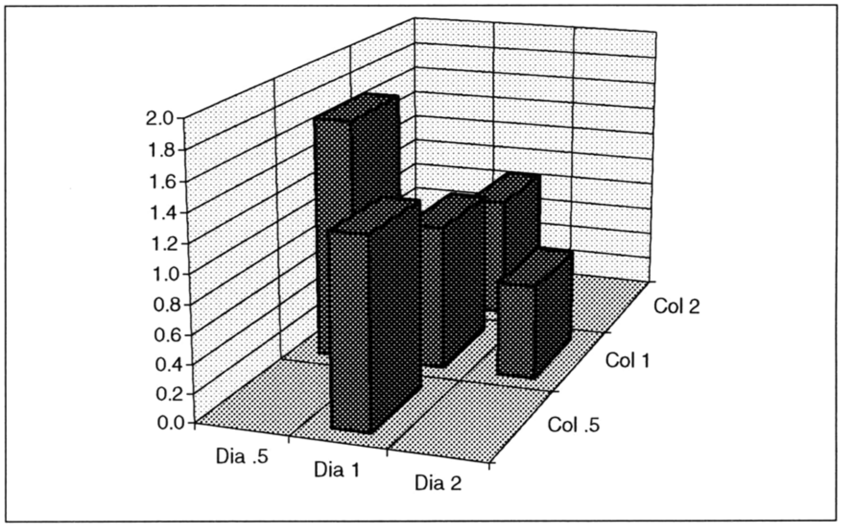
### MIT Design Method

Ratio

| | Dia .5 | Dia 1 | Dia 2 |
|---|---|---|---|
| Col .5 | | 1.306 | |
| Col 1 | 1.694 | 1.000 | 0.653 |
| Col 2 | | 0.847 | |

Deflection (ft)

| | Dia .5 | Dia 1 | Dia 2 |
|---|---|---|---|
| Col .5 | | 0.159 | |
| Col 1 | 0.206 | 0.122 | 0.079 |
| Col 2 | | 0.103 | |



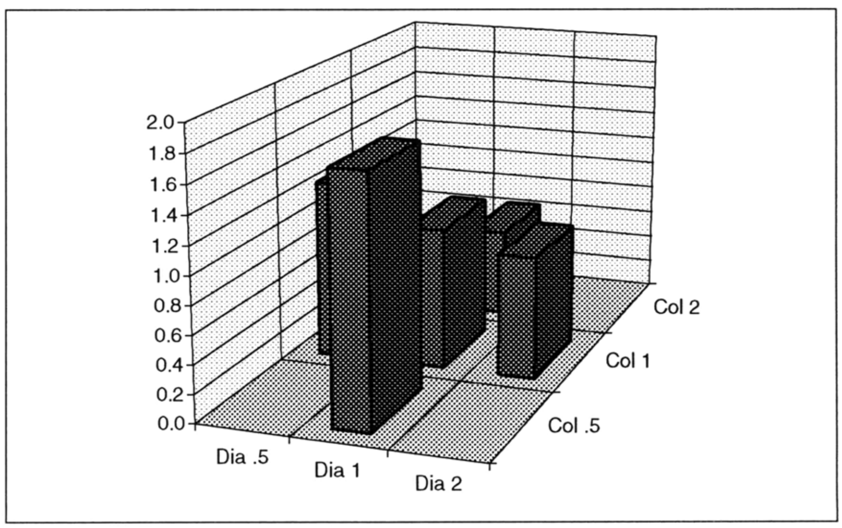

### Coupled Wall Method

Ratio

| | Dia .5 | Dia 1 | Dia 2 |
|---|---|---|---|
| Col .5 | | 1.732 | |
| Col 1 | 1.265 | 1.000 | 0.866 |
| Col 2 | | 0.633 | |

Deflection (ft)

| | Dia .5 | Dia 1 | Dia 2 |
|---|---|---|---|
| Col .5 | | 0.261 | |
| Col 1 | 0.191 | 0.151 | 0.131 |
| Col 2 | | 0.095 | |



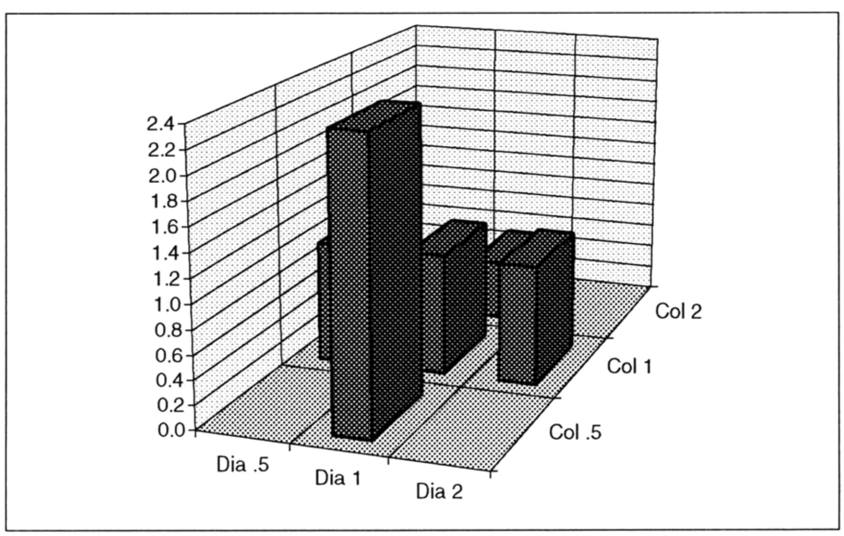

### Matlab Code

Ratio

| | Dia .5 | Dia 1 | Dia 2 |
|---|---|---|---|
| Col .5 | | 2.391 | |
| Col 1 | 1.001 | 1.000 | 0.986 |
| Col 2 | | 0.488 | |

Deflection (ft)

| | Dia .5 | Dia 1 | Dia 2 |
|---|---|---|---|
| Col .5 | | 0.581 | |
| Col 1 | 0.243 | 0.243 | 0.240 |
| Col 2 | | 0.119 | |



*Chapter 9*

# CONCLUSION

## Stafford Smith Connor Comparison / X Bracing

As seen from the results presented in the previous section, the approximation methods presented by Stafford Smith and Connor are relatively accurate. In the X - braced system, the absolute deflections computed by the MIT Design Method were accurate to within 0.000839 inches/linear foot of height representing an overall deflection error of only 0.059 inches in the 70 foot structure. For the Coupled Wall Method the absolute deflection differences were 0.012368 inches/linear foot of height and an overall structural deflection of 0.87 inches. Even though the results computed with the MIT Design Method are10 times more accurate than that calculated by the Coupled Wall Method, the estimations computed by both methods are sufficiently accurate for the initial design of a structure.

In the K - braced system, the deflection estimations were not as accurate as those computed in the X - braced frame. When Connor's MIT Design Method was applied to the K - braced frame, the absolute deflection difference was approximately 1.46, or 0.020861 inches of error per each linear foot of height. Again, the Stafford Smith Coupled Wall Method was not as accurate as the MIT Design Method. In this configuration, the deflection difference was approximately 1.11 inches for the entire structure, which translates to a 0.01584 inch deflection per linear foot of height. Although it appears that both methods are more adept at estimating deflections in an X - braced frame, the approximations calculated for a K - braced system are relatively accurate.

Although the differences between the actual and the estimated deflections ranged from 0.000839 in/ft to 0.020861 in/ft, the deflection approximations are adequate. In order to put this magnitude of error into perspective, the maximum allowable deflection for a 500 foot skyscraper is only a few inches. In short, both Connor and Stafford Smith deliver what they promise - a quick and relatively simple method for estimating deflections in both X - braced and K - braced steel structural systems that delivers accurate results.

This magnitude of accuracy is also reached when estimating the deflections for floors along the entire height of the structure. The behavior models adopted by each method assume a deflected shape. Connor assumed a combination of a quadratic and a fourth order curve in the MIT Design Method, and Stafford Smith assumed an even more complicated geometry in his Coupled Wall Method. Both performed well. From the results, one can conclude that the MIT Design Method was more accurate in approximating the absolute deflection. The error was largest when calculating deflections for floors in mid - height of the structure, but these errors tapered off as deflections were calculated for higher floors. The maximum mid - height error was less than 0.4 inches in the X - braced frame and was only 1.59 inches in the K - braced frame. In contrast, the Couple Wall Method's deflection approximation errors increased proportionately with the height of the floors. The maximum error was found when computing absolute deflections. The equations modeling the behavior of the structure are shown in Appendix IV and are best illustrated by the deflection curves presented in chapter 8.

## Deflection Comparison

The deflection behavior of both types of frames and the deflection estimations computed by both methods are best illustrated be the curves presented in the deflection comparison. The middle chart on each page of three represents the behavior of the model structure subjected to a 10 psi wind load. The charts on either side of the page show how the structure would behave if the column sizes or beam sizes were altered. These charts were added because it was interesting to see how each method dealt with the changes of member sizes. The comparison between the curves derived by each method to the actual deflection curve gives some insight into how each method works.

## Effects of Changing Diagonal and Column Sizes

The effect on the deflection estimates caused by changing the member sizes varies between all three methods. The actual behavior of the structure is most accurately modeled by the Coupled Wall Method where the absolute deflection is most affected by change in the size of the diagonals. The deflection as estimated by the MIT Design Method is affected to a larger extent by a change in the column sizes. Although the behavior is modeled more accurately by the Coupled Wall Method, the MIT Design Method is much simpler to use, and as seen in the results, the absolute deflection approximations are more accurate.

## Comparison of Methods

The advantages and disadvantages of each method are listed below:

**MIT Design Method -**

Advantages: Elegant and very simple to use. Deflection approximations are surprisingly accurate given how simple the equations are. Not only can deflections be calculated, but also the process can be reversed and member sizes can be calculated given a deflection constraint. The epitome of a "back of the envelope calculation"

Disadvantages: Modeling of the affected behavior due to the modification of member sizes is not as accurate as the Coupled Wall Method. Method is designed solely for estimating deflection in tall braced frames.

**Coupled Wall Method -**

Advantages: Although not as accurate as those found with the MIT Design Method, the deflections estimated through this procedure are sufficiently accurate for what this process was designed for. This process is very flexible - deflections for rigid frames, braced frames, and coupled walls can all be approximated using this method. The behavior of the structure is very accurately modeled using this procedure.

Disadvantages: Although the process for estimating the deflections for each type of structure is essentially the same, this procedure is slightly more complicated than the MIT Design Method. This method would prove to be an invaluable tool if programmed into a computer or calculator.

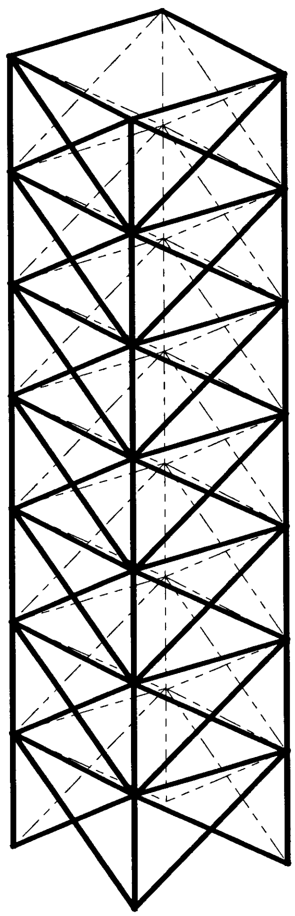
**SAP90 Structural Analysis Package -**

Advantages: The results using this finite element analysis procedure are as exact as those found using the Beam - Column Method. A graphical representation of the structural behavior is easily drawn from the calculations.

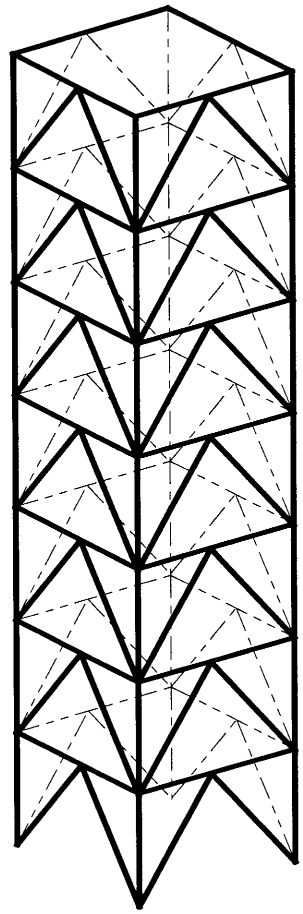
Disadvantages: The SAP90 program, when compared to the other two methods, is very demanding in terms of computational resources. The relatively large amount of time needed to build a SAP90 model makes this procedure unsuitable as a tool for use in only the preliminary design of a structure. If a model was developed at the conception of the project and used throughout the entire design process, SAP90 would be an appropriate tool to use. Having developed a structural model using this program, member size changes and load changes can be altered easily. Final design and checking of the calculations can also be accomplished with this procedure. If the design of the system layout were expected to change frequently, one of the two other approximate methods would be a more appropriate tool to use.

*Appendix I*

# TALL FRAME MODELS



**X Braced**



**K Braced**

**Structure Model**
Story Height: 10 feet
Overall Building Height: 70 feet
Building Footprint: 10 feet x 10 feet
Material Characteristics: Structural Steel (4176000 units)
Wind Load: 10 psf

*Appendix II*

## AREA CALCULATIONS

**Areas to be used by analysis**
**X Braced**

X Col = 1 Area Col = 0.01408
X Dia = 1 Area Dia = 0.00095

| | | |
|---|---|---|
| Wind Load | **b** | 0.1 |
| Height | **H** | 70 |
| Width | **B** | 10 |
| Diagonal Angle | **theta** | 45 |
| Elasticity | **E diag** | 4176000 |
| | **E Col** | 4176000 |
| Calculate | **s** | 1.16667 |
| Deflection Criteria | $\gamma *$ | 0.0025 |
| | $f *$ | 3 |
| Aspect Ratio | **B/H** | 1 to 7 |

$$D_T = \frac{bH}{\gamma *}\left[1 - \frac{x}{H}\right]$$

$$D_B = \frac{bH^3}{4s\gamma *}\left[1 - \frac{x}{H}\right]^2$$

$$D_T = A^D E^D \sin(2 \cdot theta)\cos(theta)$$

$$D_B = \frac{A^C E^C B^2}{2}$$

$$A^D = \frac{bH}{\gamma * E^D \sin(2 \cdot theta)\cos(theta)}\left[1 - \frac{x}{H}\right]$$

$$A^C = \frac{2bH^3}{4s\gamma * E^C B^2}\left[1 - \frac{x}{H}\right]^2$$

| x Location | Area of Diagonal | Area of Column |
|---|---|---|
| 70 | 0.00000 | 0.00000 |
| 68 | 0.00003 | 0.00001 |
| 66 | 0.00005 | 0.00005 |
| 64 | 0.00008 | 0.00010 |
| 62 | 0.00011 | 0.00018 |
| 60 | 0.00014 | 0.00029 |
| 58 | 0.00016 | 0.00041 |
| 56 | 0.00019 | 0.00056 |
| 54 | 0.00022 | 0.00074 |
| 52 | 0.00024 | 0.00093 |
| 50 | 0.00027 | 0.00115 |
| 48 | 0.00030 | 0.00139 |
| 46 | 0.00033 | 0.00166 |
| 44 | 0.00035 | 0.00194 |
| 42 | 0.00038 | 0.00225 |
| 40 | 0.00041 | 0.00259 |
| 38 | 0.00043 | 0.00294 |
| 36 | 0.00046 | 0.00332 |
| 34 | 0.00049 | 0.00372 |
| 32 | 0.00051 | 0.00415 |
| 30 | 0.00054 | 0.00460 |
| 28 | 0.00057 | 0.00507 |
| 26 | 0.00060 | 0.00556 |
| 24 | 0.00062 | 0.00608 |
| 22 | 0.00065 | 0.00662 |
| 20 | 0.00068 | 0.00718 |
| 18 | 0.00070 | 0.00777 |
| 16 | 0.00073 | 0.00838 |
| 14 | 0.00076 | 0.00901 |
| 12 | 0.00079 | 0.00967 |
| 10 | 0.00081 | 0.01034 |
| 8 | 0.00084 | 0.01105 |
| 6 | 0.00087 | 0.01177 |
| 4 | 0.00089 | 0.01252 |
| 2 | 0.00092 | 0.01329 |
| 0 | 0.00095 | 0.01408 |

| **Areas to be used by analysis** | K Col = 1 | Area Col = | 0.01408 |
|---|---|---|---|
| **K Braced** | K Dia = 1 | Area Dia = | 0.00393 |

| | | | | | |
|---|---|---|---|---|---|
| Wind Load | **b** | 0.1 | Diagonal Angle | **theta** | 45 |
| Height | **H** | 70 | Deflection Criteria | **s** | 1.1667 |
| Width | **B** | 10 | | $\boxed{\gamma *}$ | 0.0025 |
| Bay Height | **l** | 10 | | $\boxed{f *}$ | 3 |
| Dia Length | **L** | 14.14 | Col to Beam Ratio | **Scale** | 1 |
| Elasticity | **E beam** | 4176000 | Aspect Ratio | **B/H** | 1 to 7 |
| | **E col** | 4176000 | Calculate | **E diag** | 4176000 |

$$A^D = \frac{2L^3}{lB^2E^D\left(\left(\frac{1}{\frac{bH}{\gamma *}}\left[1-\frac{x}{H}\right]\right)-\left(\frac{l^2}{2B^2A^CE^C}\right)-\left(\frac{B(Scale)}{4E^BlA^C}\right)\right)}$$

$$A^C = \frac{2bH^3}{4s\gamma * E^C B^2}\left[1-\frac{x}{H}\right]^2$$

| x Location | Area of Diagonal | Area of Column |
|---|---|---|
| 70 | 0.00000 | 0.00000 |
| 68 | 0.00011 | 0.00001 |
| 66 | 0.00022 | 0.00005 |
| 64 | 0.00033 | 0.00010 |
| 62 | 0.00044 | 0.00018 |
| 60 | 0.00054 | 0.00029 |
| 58 | 0.00065 | 0.00041 |
| 56 | 0.00076 | 0.00056 |
| 54 | 0.00087 | 0.00074 |
| 52 | 0.00098 | 0.00093 |
| 50 | 0.00109 | 0.00115 |
| 48 | 0.00121 | 0.00139 |
| 46 | 0.00132 | 0.00166 |
| 44 | 0.00143 | 0.00194 |
| 42 | 0.00154 | 0.00225 |
| 40 | 0.00165 | 0.00259 |
| 38 | 0.00176 | 0.00294 |
| 36 | 0.00187 | 0.00332 |
| 34 | 0.00199 | 0.00372 |
| 32 | 0.00210 | 0.00415 |
| 30 | 0.00221 | 0.00460 |
| 28 | 0.00233 | 0.00507 |
| 26 | 0.00244 | 0.00556 |
| 24 | 0.00255 | 0.00608 |
| 22 | 0.00267 | 0.00662 |
| 20 | 0.00278 | 0.00718 |
| 18 | 0.00289 | 0.00777 |
| 16 | 0.00301 | 0.00838 |
| 14 | 0.00312 | 0.00901 |
| 12 | 0.00324 | 0.00967 |
| 10 | 0.00335 | 0.01034 |
| 8 | 0.00347 | 0.01105 |
| 6 | 0.00358 | 0.01177 |
| 4 | 0.00370 | 0.01252 |
| 2 | 0.00382 | 0.01329 |
| 0 | 0.00393 | 0.01408 |

## *Appendix III*

## MEMBER SIZES

## Member Sizes

Area of members used in each analysis scheme

### K Braced Frame

| | Excel<br>square feet | SAP90<br>square feet | Side Length<br>feet |
|---|---|---|---|
| **Col 1** | 0.0141 | 0.0070 | 0.0839 |
| **Beam 1** | 0.0141 | 0.0070 | 0.0839 |
| **Dia 1** | 0.0039 | 0.0020 | 0.0443 |
| **Col 1** | 0.0141 | 0.0070 | 0.0839 |
| **Beam 1** | 0.0141 | 0.0070 | 0.0839 |
| **Dia .5** | 0.0020 | 0.0010 | 0.0313 |
| **Col 1** | 0.0141 | 0.0070 | 0.0839 |
| **Beam 1** | 0.0141 | 0.0070 | 0.0839 |
| **Dia 2** | 0.0079 | 0.0039 | 0.0627 |
| **Col .5** | 0.0070 | 0.0035 | 0.0593 |
| **Beam 1** | 0.0141 | 0.0070 | 0.0839 |
| **Dia 1** | 0.0039 | 0.0020 | 0.0443 |
| **Col 2** | 0.0282 | 0.0141 | 0.1187 |
| **Beam 1** | 0.0141 | 0.0070 | 0.0839 |
| **Dia 1** | 0.0039 | 0.0020 | 0.0443 |

### X Braced Frame

| | Excel<br>square feet | SAP90<br>square feet | Side Length<br>feet |
|---|---|---|---|
| **Col 1** | 0.0141 | 0.0070 | 0.0839 |
| **Dia 1** | 0.0010 | 0.0005 | 0.0218 |
| **Col 1** | 0.0141 | 0.0070 | 0.0839 |
| **Dia .5** | 0.0005 | 0.0002 | 0.0154 |
| **Col 1** | 0.0141 | 0.0070 | 0.0839 |
| **Dia 2** | 0.0019 | 0.0010 | 0.0308 |
| **Col .5** | 0.0070 | 0.0035 | 0.0593 |
| **Dia 1** | 0.0010 | 0.0005 | 0.0218 |
| **Col 2** | 0.0282 | 0.0141 | 0.1187 |
| **Dia 1** | 0.0010 | 0.0005 | 0.0218 |

## *Appendix IV*

# DEFLECTION CALCULATIONS

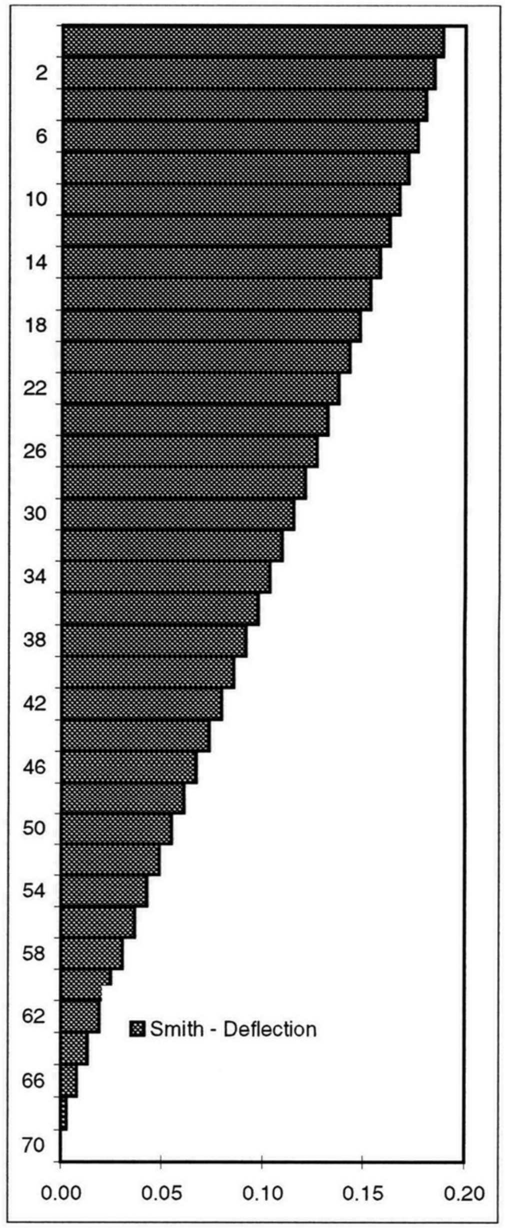
*Calculations* *Page*

| | | |
|---|---|---|
| w | Uniform Load | 0.1 |
| H | Structure Height | 70 |
| h | Story Height | 10 |
| l | Frame Width | 10 |
| Ac | Area Col | 0.01408 |
| Ad | Area Dia | 0.00095 |
| E | Modulus | 4176000 |
| I | (Calculated) | 0.00070 |
| Ig | (Calculated) | 0.70473 |
| k | (Calculated) | 1.00050 |

$$\boxed{GA = \frac{2hl^2E}{\left[\dfrac{h^3}{A_c} + \dfrac{(l^2 + h^2)^{3/2}}{A_d}\right]}}$$

| | |
|---|---|
| GA = | 2734.88372 |
| EI = | 2940.00000 |
| alpha = | 0.96449 |

| x location | Deflection (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0029 |
| 66 | 0.0080 |
| 64 | 0.0134 |
| 62 | 0.0191 |
| 60 | 0.0249 |
| 58 | 0.0307 |
| 56 | 0.0367 |
| 54 | 0.0427 |
| 52 | 0.0487 |
| 50 | 0.0548 |
| 48 | 0.0610 |
| 46 | 0.0671 |
| 44 | 0.0733 |
| 42 | 0.0794 |
| 40 | 0.0855 |
| 38 | 0.0915 |
| 36 | 0.0975 |
| 34 | 0.1035 |
| 32 | 0.1094 |
| 30 | 0.1152 |
| 28 | 0.1209 |
| 26 | 0.1265 |
| 24 | 0.1321 |
| 22 | 0.1375 |
| 20 | 0.1428 |
| 18 | 0.1480 |
| 16 | 0.1531 |
| 14 | 0.1580 |
| 12 | 0.1628 |
| 10 | 0.1675 |
| 8 | 0.1720 |
| 6 | 0.1764 |
| 4 | 0.1807 |
| 2 | 0.1848 |
| 0 | 0.1888 |

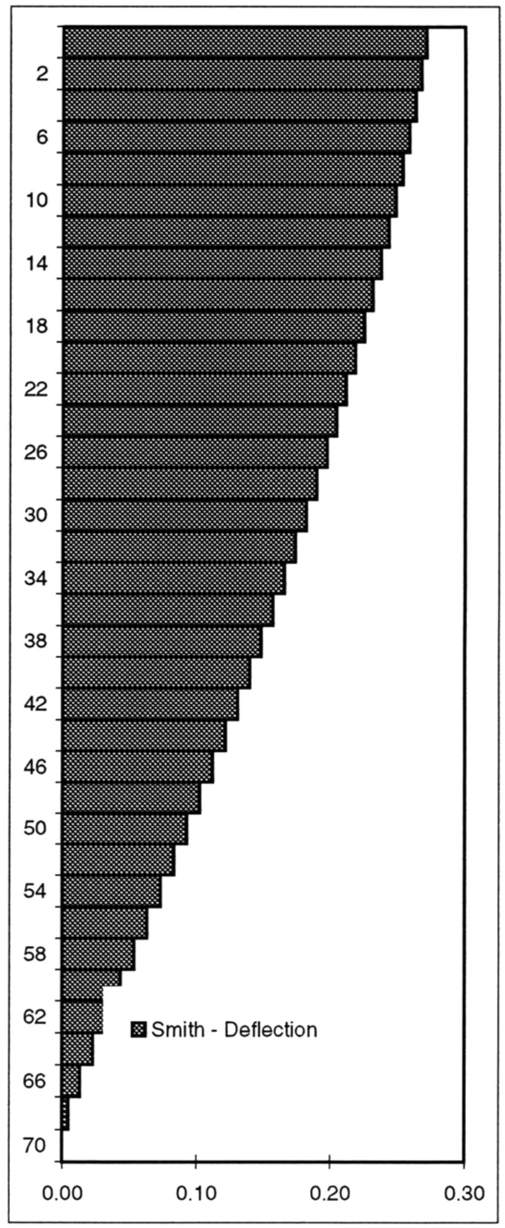


| | | |
|---|---|---|
| w | Uniform Load | 0.1 |
| H | Structure Height | 70 |
| h | Story Height | 10 |
| l | Frame Width | 10 |
| Ac | Area Col | 0.01408 |
| Ad | Area Dia | 0.00047 |
| E | Modulus | 4176000 |
| I | (Calculated) | 0.00070 |
| Ig | (Calculated) | 0.70473 |
| k | (Calculated) | 1.00050 |

$$\boxed{GA = \frac{2hl^2E}{\left[\dfrac{h^3}{A_c} + \dfrac{(l^2+h^2)^{3/2}}{A_d}\right]}}$$

| | |
|---|---|
| GA = | 1383.52941 |
| EI = | 2940.00000 |
| alpha = | 0.68599 |

| x location | Deflection (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0046 |
| 66 | 0.0134 |
| 64 | 0.0232 |
| 62 | 0.0332 |
| 60 | 0.0433 |
| 58 | 0.0534 |
| 56 | 0.0634 |
| 54 | 0.0733 |
| 52 | 0.0832 |
| 50 | 0.0929 |
| 48 | 0.1025 |
| 46 | 0.1120 |
| 44 | 0.1214 |
| 42 | 0.1306 |
| 40 | 0.1396 |
| 38 | 0.1484 |
| 36 | 0.1571 |
| 34 | 0.1655 |
| 32 | 0.1737 |
| 30 | 0.1818 |
| 28 | 0.1896 |
| 26 | 0.1971 |
| 24 | 0.2044 |
| 22 | 0.2115 |
| 20 | 0.2183 |
| 18 | 0.2249 |
| 16 | 0.2311 |
| 14 | 0.2371 |
| 12 | 0.2429 |
| 10 | 0.2483 |
| 8 | 0.2535 |
| 6 | 0.2584 |
| 4 | 0.2630 |
| 2 | 0.2674 |
| 0 | 0.2715 |

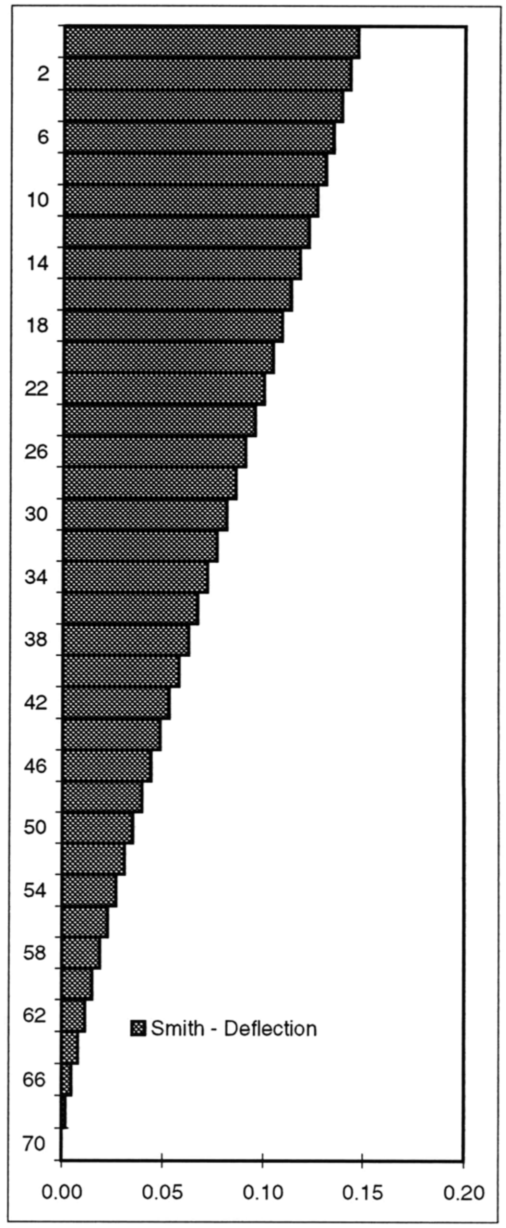


| | | |
|---|---|---|
| w | Uniform Load | 0.1 |
| H | Structure Height | 70 |
| h | Story Height | 10 |
| l | Frame Width | 10 |
| Ac | Area Col | 0.01408 |
| Ad | Area Dia | 0.00190 |
| E | Modulus | 4176000 |
| I | (Calculated) | 0.00070 |
| Ig | (Calculated) | 0.70473 |
| k | (Calculated) | 1.00050 |

$$\boxed{GA = \frac{2hl^2E}{\left[\frac{h^3}{A_c} + \frac{(l^2+h^2)^{3/2}}{A_d}\right]}}$$

| | |
|---|---|
| GA = | 5345.45455 |
| EI = | 2940.00000 |
| alpha = | 1.34840 |

| x location | Deflection (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0018 |
| 66 | 0.0048 |
| 64 | 0.0080 |
| 62 | 0.0114 |
| 60 | 0.0149 |
| 58 | 0.0187 |
| 56 | 0.0226 |
| 54 | 0.0267 |
| 52 | 0.0309 |
| 50 | 0.0351 |
| 48 | 0.0395 |
| 46 | 0.0440 |
| 44 | 0.0485 |
| 42 | 0.0531 |
| 40 | 0.0577 |
| 38 | 0.0624 |
| 36 | 0.0671 |
| 34 | 0.0718 |
| 32 | 0.0765 |
| 30 | 0.0812 |
| 28 | 0.0859 |
| 26 | 0.0906 |
| 24 | 0.0952 |
| 22 | 0.0998 |
| 20 | 0.1044 |
| 18 | 0.1089 |
| 16 | 0.1134 |
| 14 | 0.1178 |
| 12 | 0.1221 |
| 10 | 0.1264 |
| 8 | 0.1306 |
| 6 | 0.1348 |
| 4 | 0.1388 |
| 2 | 0.1428 |
| 0 | 0.1468 |

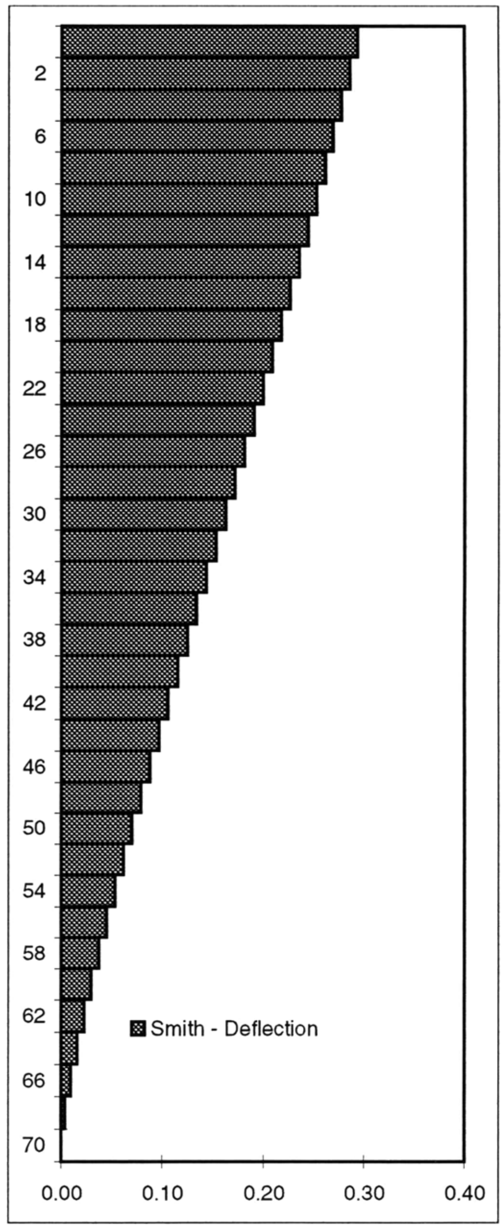


| | | |
|---|---|---|
| w | Uniform Load | 0.1 |
| H | Structure Height | 70 |
| h | Story Height | 10 |
| l | Frame Width | 10 |
| Ac | Area Col | 0.00704 |
| Ad | Area Dia | 0.00095 |
| E | Modulus | 4176000 |
| I | (Calculated) | 0.00035 |
| Ig | (Calculated) | 0.35236 |
| k | (Calculated) | 1.00050 |

$$\boxed{GA = \frac{2hl^2E}{\left[\dfrac{h^3}{A_c} + \dfrac{(l^2+h^2)^{3/2}}{A_d}\right]}}$$

| | |
|---|---|
| GA = | 2672.72727 |
| EI = | 1470.00000 |
| alpha = | 1.34840 |

| x location | Deflection (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0037 |
| 66 | 0.0095 |
| 64 | 0.0159 |
| 62 | 0.0227 |
| 60 | 0.0299 |
| 58 | 0.0374 |
| 56 | 0.0452 |
| 54 | 0.0534 |
| 52 | 0.0617 |
| 50 | 0.0703 |
| 48 | 0.0790 |
| 46 | 0.0880 |
| 44 | 0.0970 |
| 42 | 0.1062 |
| 40 | 0.1155 |
| 38 | 0.1248 |
| 36 | 0.1342 |
| 34 | 0.1436 |
| 32 | 0.1530 |
| 30 | 0.1624 |
| 28 | 0.1718 |
| 26 | 0.1812 |
| 24 | 0.1905 |
| 22 | 0.1997 |
| 20 | 0.2088 |
| 18 | 0.2178 |
| 16 | 0.2268 |
| 14 | 0.2356 |
| 12 | 0.2443 |
| 10 | 0.2528 |
| 8 | 0.2612 |
| 6 | 0.2695 |
| 4 | 0.2777 |
| 2 | 0.2857 |
| 0 | 0.2935 |

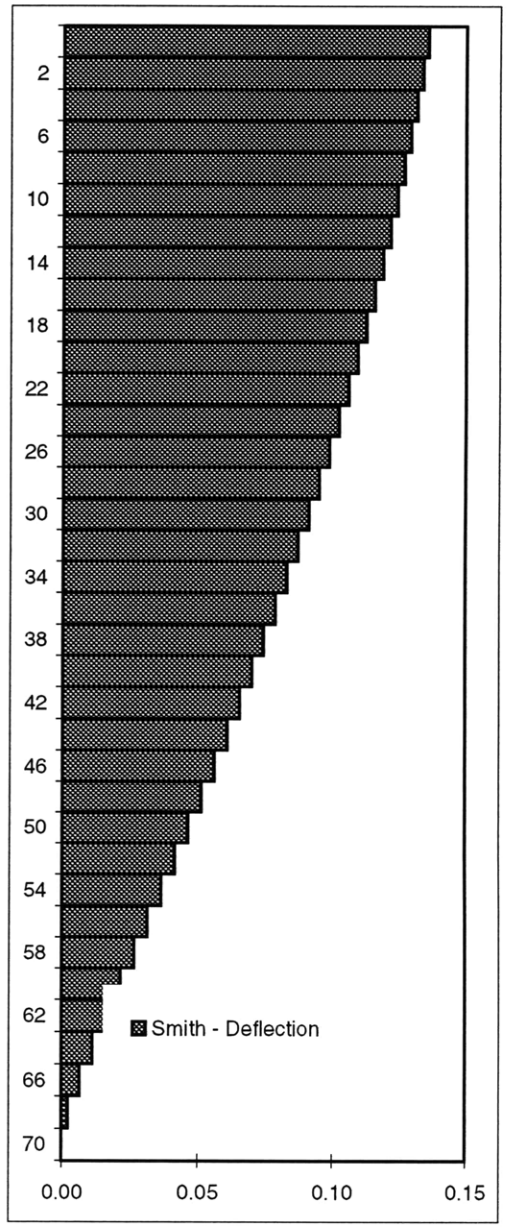


| | | |
|---|---|---|
| w | Uniform Load | 0.1 |
| H | Structure Height | 70 |
| h | Story Height | 10 |
| l | Frame Width | 10 |
| Ac | Area Col | 0.02816 |
| Ad | Area Dia | 0.00095 |
| E | Modulus | 4176000 |
| I | (Calculated) | 0.00141 |
| Ig | (Calculated) | 1.40945 |
| k | (Calculated) | 1.00050 |

$$\boxed{GA = \frac{2hl^2E}{\left[\dfrac{h^3}{A_c} + \dfrac{(l^2+h^2)^{3/2}}{A_d}\right]}}$$

| | |
|---|---|
| GA = | 2767.05882 |
| EI = | 5880.00000 |
| alpha = | 0.68599 |

| x location | Deflection (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0023 |
| 66 | 0.0067 |
| 64 | 0.0116 |
| 62 | 0.0166 |
| 60 | 0.0217 |
| 58 | 0.0267 |
| 56 | 0.0317 |
| 54 | 0.0367 |
| 52 | 0.0416 |
| 50 | 0.0465 |
| 48 | 0.0513 |
| 46 | 0.0560 |
| 44 | 0.0607 |
| 42 | 0.0653 |
| 40 | 0.0698 |
| 38 | 0.0742 |
| 36 | 0.0785 |
| 34 | 0.0828 |
| 32 | 0.0869 |
| 30 | 0.0909 |
| 28 | 0.0948 |
| 26 | 0.0986 |
| 24 | 0.1022 |
| 22 | 0.1057 |
| 20 | 0.1092 |
| 18 | 0.1124 |
| 16 | 0.1156 |
| 14 | 0.1186 |
| 12 | 0.1214 |
| 10 | 0.1242 |
| 8 | 0.1268 |
| 6 | 0.1292 |
| 4 | 0.1315 |
| 2 | 0.1337 |
| 0 | 0.1358 |



| | | |
|---|---|---|
| Wind Load | **b** | 0.1 |
| Height | **H** | 70 |
| Width | **B** | 10 |
| Dia Angle | **theta** | 45 |
| Elasticity | **E diag** | 4176000 |
| | **E col** | 4176000 |
| Area | **A col** | 0.01408 |
| | **A diag** | 0.00095 |
| Calculate | **s** | 1.16667 |
| | $\boxed{f*}$ | 3 |

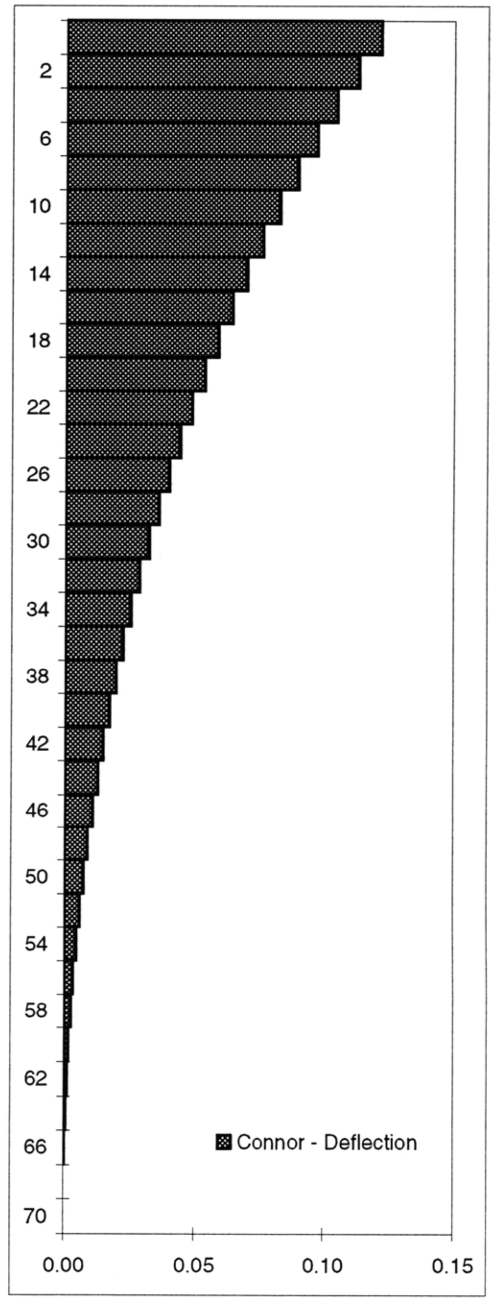
$$\boxed{Deflection = \frac{b(H-x)^4}{24D_B} + \frac{b(H-x)^2}{2D_T}}$$

$$\boxed{D_B = \frac{A^C E^C B^2}{2}}$$

$$\boxed{D_T = A^D E^D \sin(2\cdot theta)\cos(theta)}$$

Asp. Ratio **B/H** 1 to 7

| x Location | Deflection (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0001 |
| 66 | 0.0003 |
| 64 | 0.0006 |
| 62 | 0.0011 |
| 60 | 0.0018 |
| 58 | 0.0026 |
| 56 | 0.0036 |
| 54 | 0.0047 |
| 52 | 0.0059 |
| 50 | 0.0074 |
| 48 | 0.0090 |
| 46 | 0.0108 |
| 44 | 0.0127 |
| 42 | 0.0149 |
| 40 | 0.0172 |
| 38 | 0.0198 |
| 36 | 0.0225 |
| 34 | 0.0255 |
| 32 | 0.0287 |
| 30 | 0.0322 |
| 28 | 0.0359 |
| 26 | 0.0399 |
| 24 | 0.0441 |
| 22 | 0.0487 |
| 20 | 0.0535 |
| 18 | 0.0586 |
| 16 | 0.0641 |
| 14 | 0.0699 |
| 12 | 0.0761 |
| 10 | 0.0827 |
| 8 | 0.0896 |
| 6 | 0.0969 |
| 4 | 0.1047 |
| 2 | 0.1129 |
| 0 | 0.1215 |



| | | |
|---|---|---|
| Wind Load | **b** | 0.1 |
| Height | **H** | 70 |
| Width | **B** | 10 |
| Dia Angle | **theta** | 45 |
| Elasticity | **E diag** | 4176000 |
| | **E col** | 4176000 |
| Area | **A col** | 0.01408 |
| | **A diag** | 0.00047 |
| Calculate | **s** | 1.16667 |
| | $\boxed{f*}$ | 3 |

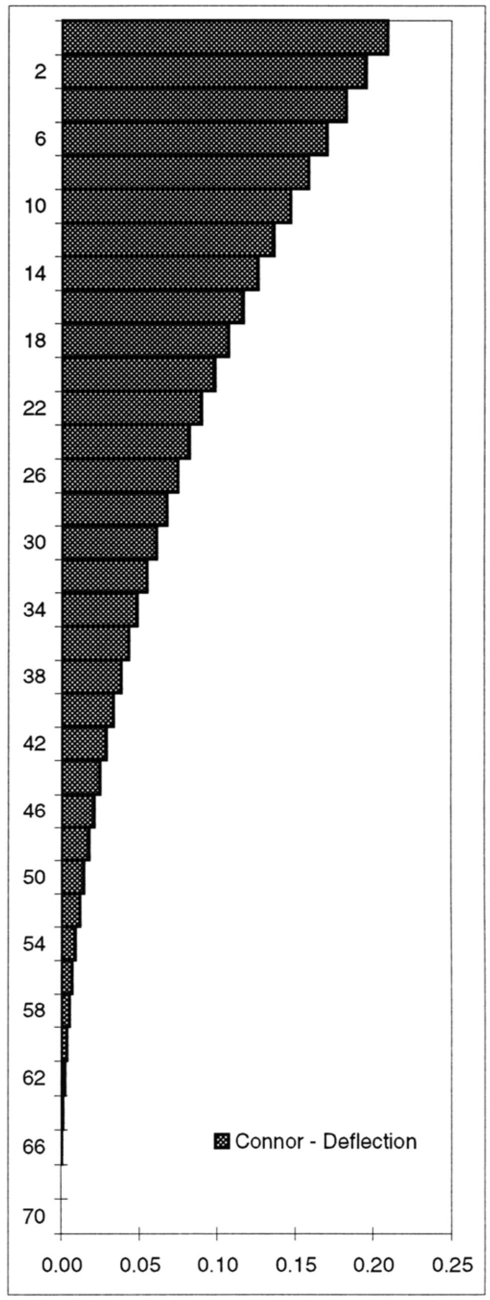
$$\boxed{Deflection = \frac{b(H-x)^4}{24D_B} + \frac{b(H-x)^2}{2D_T}}$$

$$\boxed{D_B = \frac{A^C E^C B^2}{2}}$$

$$\boxed{D_T = A^D E^D \sin(2 \cdot theta)\cos(theta)}$$

Asp. Ratio **B/H** 1 to 7

| x Location | Deflection (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0001 |
| 66 | 0.0006 |
| 64 | 0.0013 |
| 62 | 0.0023 |
| 60 | 0.0036 |
| 58 | 0.0052 |
| 56 | 0.0071 |
| 54 | 0.0092 |
| 52 | 0.0117 |
| 50 | 0.0145 |
| 48 | 0.0176 |
| 46 | 0.0210 |
| 44 | 0.0248 |
| 42 | 0.0289 |
| 40 | 0.0333 |
| 38 | 0.0381 |
| 36 | 0.0432 |
| 34 | 0.0487 |
| 32 | 0.0545 |
| 30 | 0.0608 |
| 28 | 0.0674 |
| 26 | 0.0745 |
| 24 | 0.0819 |
| 22 | 0.0898 |
| 20 | 0.0981 |
| 18 | 0.1069 |
| 16 | 0.1162 |
| 14 | 0.1259 |
| 12 | 0.1362 |
| 10 | 0.1469 |
| 8 | 0.1582 |
| 6 | 0.1701 |
| 4 | 0.1825 |
| 2 | 0.1954 |
| 0 | 0.2090 |



| | | |
|---|---|---|
| Wind Load | **b** | 0.1 |
| Height | **H** | 70 |
| Width | **B** | 10 |
| Dia Angle | **theta** | 45 |
| Elasticity | **E diag** | 4176000 |
| | **E col** | 4176000 |
| Area | **A col** | 0.01408 |
| | **A diag** | 0.00190 |
| Calculate | **s** | 1.16667 |
| | $\boxed{f*}$ | 3 |

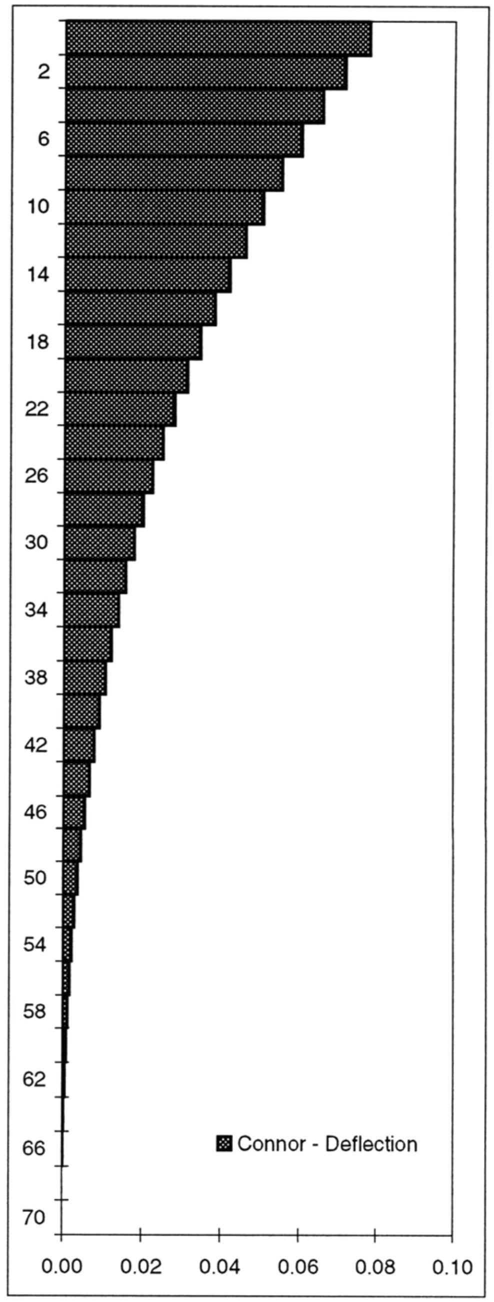
$$\boxed{Deflection = \frac{b(H-x)^4}{24 D_B} + \frac{b(H-x)^2}{2 D_T}}$$

$$\boxed{D_B = \frac{A^C E^C B^2}{2}}$$

$$\boxed{D_T = A^D E^D \sin(2 \cdot theta)\cos(theta)}$$

Asp. Ratio **B/H** 1 to 7

| x Location | Deflection (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0000 |
| 66 | 0.0001 |
| 64 | 0.0003 |
| 62 | 0.0006 |
| 60 | 0.0009 |
| 58 | 0.0013 |
| 56 | 0.0018 |
| 54 | 0.0024 |
| 52 | 0.0030 |
| 50 | 0.0038 |
| 48 | 0.0047 |
| 46 | 0.0056 |
| 44 | 0.0067 |
| 42 | 0.0079 |
| 40 | 0.0092 |
| 38 | 0.0106 |
| 36 | 0.0122 |
| 34 | 0.0140 |
| 32 | 0.0158 |
| 30 | 0.0179 |
| 28 | 0.0202 |
| 26 | 0.0226 |
| 24 | 0.0252 |
| 22 | 0.0281 |
| 20 | 0.0312 |
| 18 | 0.0345 |
| 16 | 0.0381 |
| 14 | 0.0419 |
| 12 | 0.0461 |
| 10 | 0.0505 |
| 8 | 0.0553 |
| 6 | 0.0603 |
| 4 | 0.0658 |
| 2 | 0.0716 |
| 0 | 0.0778 |



| | | |
|---|---|---|
| Wind Load | **b** | 0.1 |
| Height | **H** | 70 |
| Width | **B** | 10 |
| Dia Angle | **theta** | 45 |
| Elasticity | **E diag** | 4176000 |
| | **E col** | 4176000 |
| Area | **A col** | 0.00704 |
| | **A diag** | 0.00095 |
| Calculate | **s** | 1.16667 |
| | $\boxed{f*}$ | 3 |

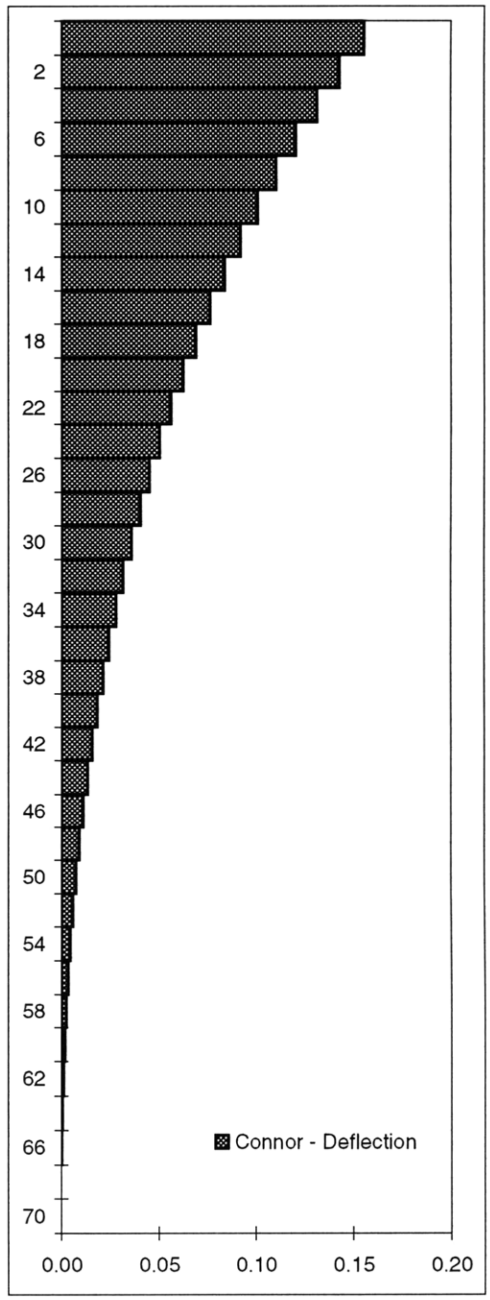
$$\boxed{Deflection = \frac{b(H-x)^4}{24 D_B} + \frac{b(H-x)^2}{2 D_T}}$$

$$\boxed{D_B = \frac{A^C E^C B^2}{2}}$$

$$\boxed{D_T = A^D E^D \sin(2 \cdot theta)\cos(theta)}$$

Asp. Ratio **B/H** 1 to 7

| x Location | Deflection (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0001 |
| 66 | 0.0003 |
| 64 | 0.0006 |
| 62 | 0.0012 |
| 60 | 0.0018 |
| 58 | 0.0026 |
| 56 | 0.0036 |
| 54 | 0.0048 |
| 52 | 0.0061 |
| 50 | 0.0076 |
| 48 | 0.0093 |
| 46 | 0.0112 |
| 44 | 0.0134 |
| 42 | 0.0157 |
| 40 | 0.0184 |
| 38 | 0.0213 |
| 36 | 0.0244 |
| 34 | 0.0279 |
| 32 | 0.0317 |
| 30 | 0.0358 |
| 28 | 0.0403 |
| 26 | 0.0452 |
| 24 | 0.0505 |
| 22 | 0.0562 |
| 20 | 0.0624 |
| 18 | 0.0690 |
| 16 | 0.0762 |
| 14 | 0.0839 |
| 12 | 0.0921 |
| 10 | 0.1010 |
| 8 | 0.1105 |
| 6 | 0.1207 |
| 4 | 0.1316 |
| 2 | 0.1432 |
| 0 | 0.1556 |



| | | |
|---|---|---|
| Wind Load | **b** | 0.1 |
| Height | **H** | 70 |
| Width | **B** | 10 |
| Dia Angle | **theta** | 45 |
| Elasticity | **E diag** | 4176000 |
| | **E col** | 4176000 |
| Area | **A col** | 0.02816 |
| | **A diag** | 0.00095 |
| Calculate | **s** | 1.16667 |
| | $\boxed{f*}$ | 3 |

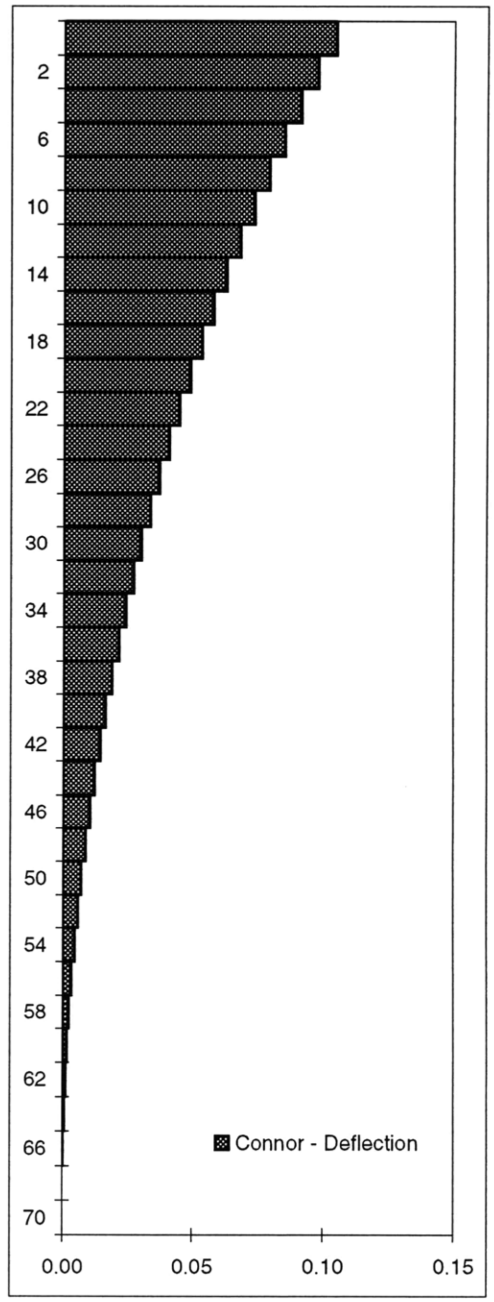
$$\boxed{Deflection = \frac{b(H-x)^4}{24D_B} + \frac{b(H-x)^2}{2D_T}}$$

$$\boxed{D_B = \frac{A^C E^C B^2}{2}}$$

$$\boxed{D_T = A^D E^D \sin(2 \cdot theta)\cos(theta)}$$

Asp. Ratio **B/H** 1 to 7

| x Location | Deflection (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0001 |
| 66 | 0.0003 |
| 64 | 0.0006 |
| 62 | 0.0011 |
| 60 | 0.0018 |
| 58 | 0.0026 |
| 56 | 0.0035 |
| 54 | 0.0046 |
| 52 | 0.0059 |
| 50 | 0.0073 |
| 48 | 0.0088 |
| 46 | 0.0105 |
| 44 | 0.0124 |
| 42 | 0.0144 |
| 40 | 0.0166 |
| 38 | 0.0190 |
| 36 | 0.0216 |
| 34 | 0.0243 |
| 32 | 0.0273 |
| 30 | 0.0304 |
| 28 | 0.0337 |
| 26 | 0.0372 |
| 24 | 0.0410 |
| 22 | 0.0449 |
| 20 | 0.0491 |
| 18 | 0.0535 |
| 16 | 0.0581 |
| 14 | 0.0630 |
| 12 | 0.0681 |
| 10 | 0.0735 |
| 8 | 0.0791 |
| 6 | 0.0850 |
| 4 | 0.0912 |
| 2 | 0.0977 |
| 0 | 0.1045 |

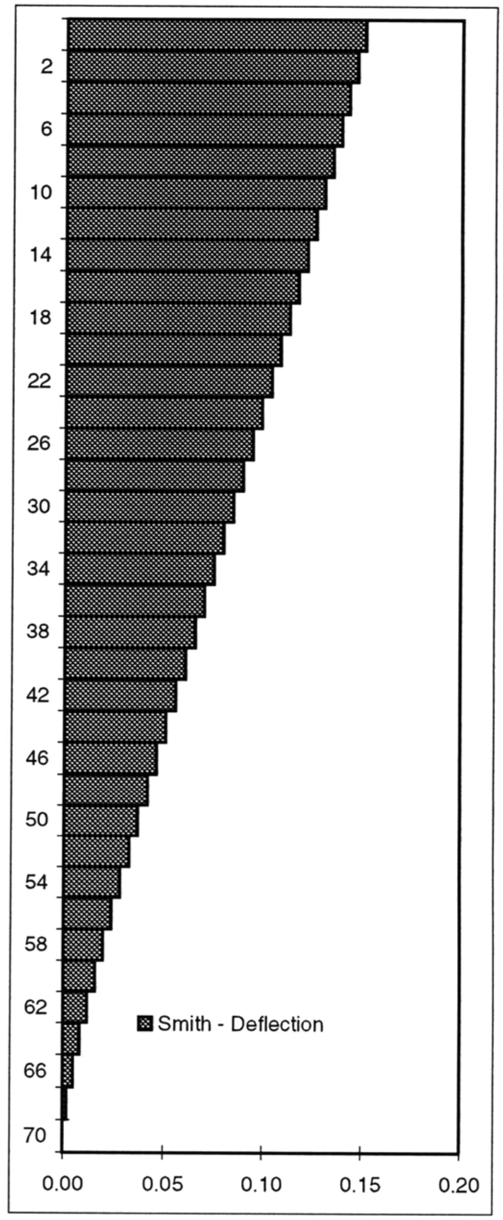


| | | |
|---|---|---|
| w | Uniform Load | 0.1 |
| H | Structure Height | 70 |
| h | Story Height | 10 |
| l | Frame Width | 10 |
| Ac | Area Col | 0.01408 |
| Ad | Area Dia | 0.00393 |
| E | Modulus | 4176000 |
| I | (Calculated) | 0.00070 |
| Ig | (Calculated) | 0.70473 |
| k | (Calculated) | 1.00050 |

$$GA = \frac{hl^2 E}{2\left[\frac{h^3}{A_c} + \frac{[(l/2)^2 + h^2]^{3/2}}{A_d}\right]}$$

| | |
|---|---|
| GA = | 4897.69639 |
| EI = | 2940.00000 |
| alpha = | 1.29069 |

| x location | Deflection (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0020 |
| 66 | 0.0051 |
| 64 | 0.0085 |
| 62 | 0.0121 |
| 60 | 0.0159 |
| 58 | 0.0199 |
| 56 | 0.0240 |
| 54 | 0.0282 |
| 52 | 0.0326 |
| 50 | 0.0371 |
| 48 | 0.0416 |
| 46 | 0.0462 |
| 44 | 0.0509 |
| 42 | 0.0556 |
| 40 | 0.0604 |
| 38 | 0.0652 |
| 36 | 0.0700 |
| 34 | 0.0749 |
| 32 | 0.0797 |
| 30 | 0.0845 |
| 28 | 0.0893 |
| 26 | 0.0941 |
| 24 | 0.0988 |
| 22 | 0.1035 |
| 20 | 0.1081 |
| 18 | 0.1127 |
| 16 | 0.1172 |
| 14 | 0.1217 |
| 12 | 0.1261 |
| 10 | 0.1304 |
| 8 | 0.1346 |
| 6 | 0.1388 |
| 4 | 0.1429 |
| 2 | 0.1469 |
| 0 | 0.1508 |

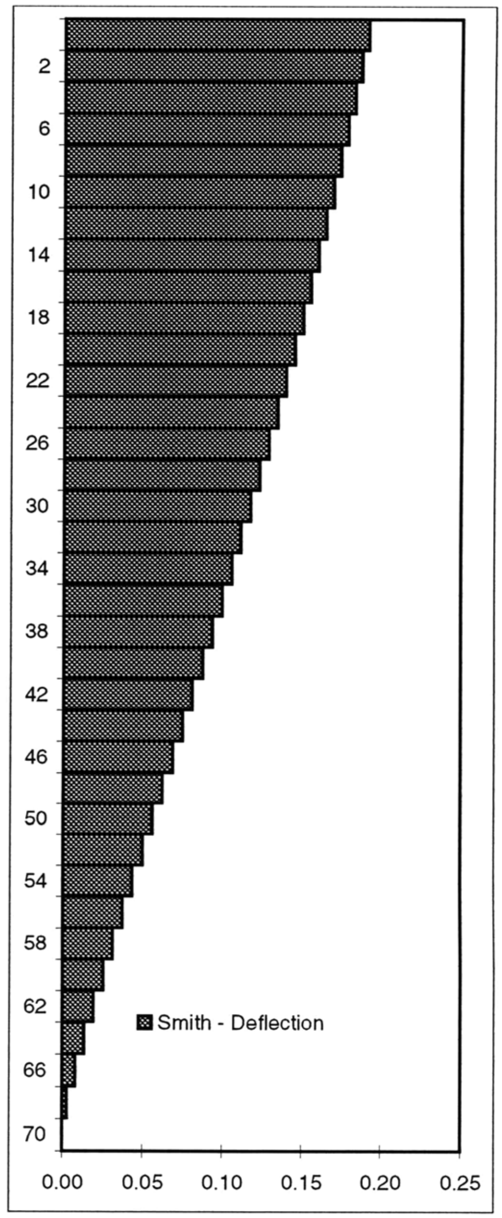


| | | |
|---|---|---|
| w | Uniform Load | 0.1 |
| H | Structure Height | 70 |
| h | Story Height | 10 |
| l | Frame Width | 10 |
| Ac | Area Col | 0.01408 |
| Ad | Area Dia | 0.00197 |
| E | Modulus | 4176000 |
| I | (Calculated) | 0.00070 |
| Ig | (Calculated) | 0.70473 |
| k | (Calculated) | 1.00050 |

$$\boxed{GA = \frac{hl^2 E}{2\left[\dfrac{h^3}{A_c} + \dfrac{[(l/2)^2 + h^2]^{3/2}}{A_d}\right]}}$$

| | |
|---|---|
| GA = | 2671.35659 |
| EI = | 2940.00000 |
| alpha = | 0.95322 |

| x location | Deflection (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0030 |
| 66 | 0.0081 |
| 64 | 0.0137 |
| 62 | 0.0195 |
| 60 | 0.0253 |
| 58 | 0.0313 |
| 56 | 0.0373 |
| 54 | 0.0434 |
| 52 | 0.0496 |
| 50 | 0.0558 |
| 48 | 0.0620 |
| 46 | 0.0682 |
| 44 | 0.0744 |
| 42 | 0.0806 |
| 40 | 0.0868 |
| 38 | 0.0929 |
| 36 | 0.0990 |
| 34 | 0.1050 |
| 32 | 0.1110 |
| 30 | 0.1168 |
| 28 | 0.1226 |
| 26 | 0.1283 |
| 24 | 0.1339 |
| 22 | 0.1393 |
| 20 | 0.1447 |
| 18 | 0.1499 |
| 16 | 0.1550 |
| 14 | 0.1600 |
| 12 | 0.1648 |
| 10 | 0.1695 |
| 8 | 0.1740 |
| 6 | 0.1784 |
| 4 | 0.1827 |
| 2 | 0.1868 |
| 0 | 0.1908 |

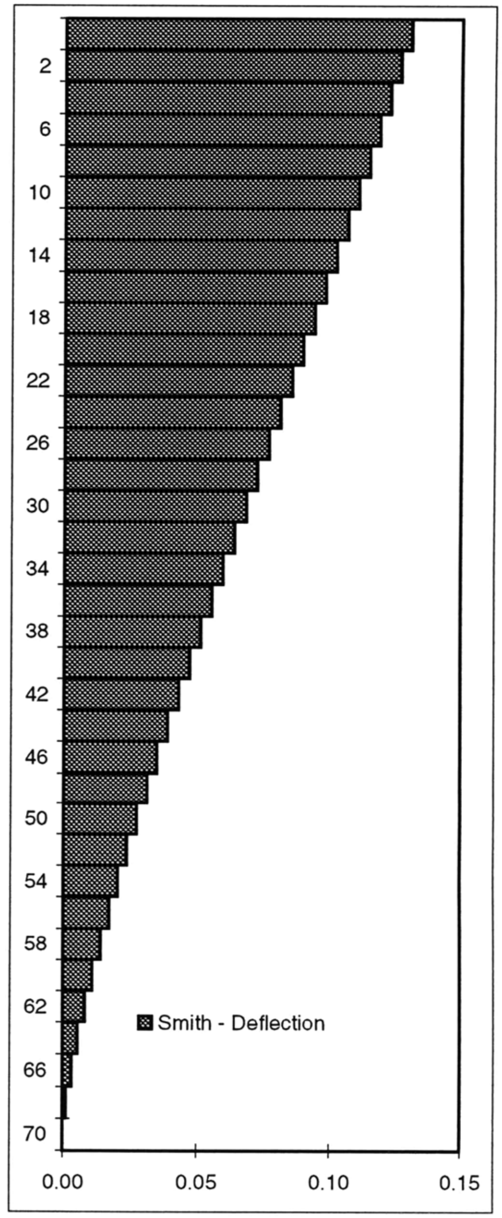


| | | |
|---|---|---|
| w | Uniform Load | 0.1 |
| H | Structure Height | 70 |
| h | Story Height | 10 |
| l | Frame Width | 10 |
| Ac | Area Col | 0.01408 |
| Ad | Area Dia | 0.00787 |
| E | Modulus | 4176000 |
| I | (Calculated) | 0.00070 |
| Ig | (Calculated) | 0.70473 |
| k | (Calculated) | 1.00050 |

$$\boxed{GA = \frac{hl^2E}{2\left[\dfrac{h^3}{A_c} + \dfrac{[(l/2)^2 + h^2]^{3/2}}{A_d}\right]}}$$

| | |
|---|---|
| GA = | 8396.61487 |
| EI = | 2940.00000 |
| alpha = | 1.68997 |

| x location | Deflection (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0013 |
| 66 | 0.0034 |
| 64 | 0.0057 |
| 62 | 0.0083 |
| 60 | 0.0110 |
| 58 | 0.0140 |
| 56 | 0.0171 |
| 54 | 0.0204 |
| 52 | 0.0239 |
| 50 | 0.0275 |
| 48 | 0.0312 |
| 46 | 0.0350 |
| 44 | 0.0389 |
| 42 | 0.0429 |
| 40 | 0.0470 |
| 38 | 0.0512 |
| 36 | 0.0554 |
| 34 | 0.0596 |
| 32 | 0.0638 |
| 30 | 0.0681 |
| 28 | 0.0724 |
| 26 | 0.0767 |
| 24 | 0.0810 |
| 22 | 0.0853 |
| 20 | 0.0896 |
| 18 | 0.0939 |
| 16 | 0.0981 |
| 14 | 0.1023 |
| 12 | 0.1065 |
| 10 | 0.1106 |
| 8 | 0.1147 |
| 6 | 0.1187 |
| 4 | 0.1227 |
| 2 | 0.1267 |
| 0 | 0.1306 |

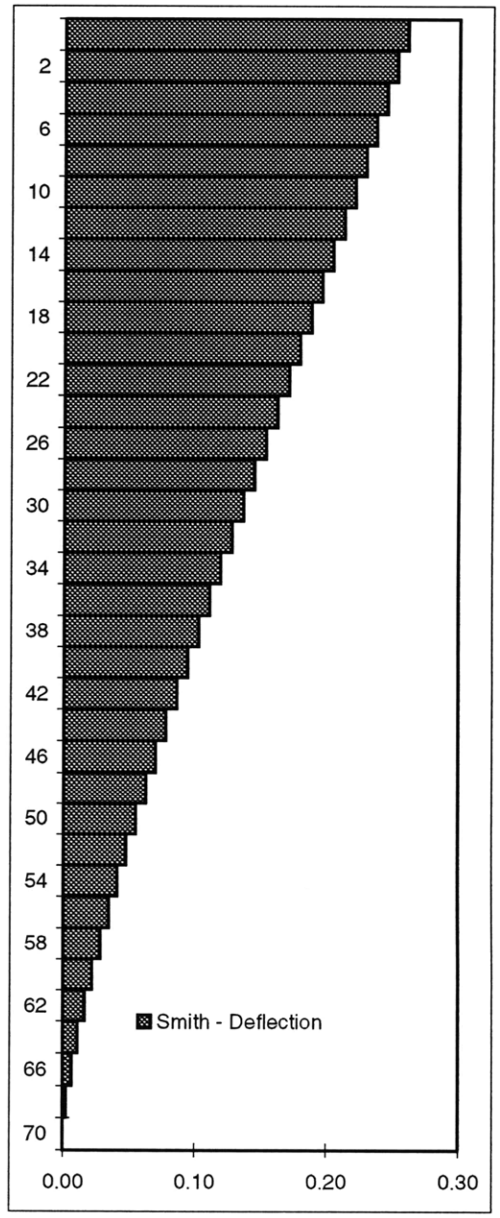


| | | |
|---|---|---|
| w | Uniform Load | 0.1 |
| H | Structure Height | 70 |
| h | Story Height | 10 |
| l | Frame Width | 10 |
| Ac | Area Col | 0.00704 |
| Ad | Area Dia | 0.00393 |
| E | Modulus | 4176000 |
| I | (Calculated) | 0.00035 |
| Ig | (Calculated) | 0.35236 |
| k | (Calculated) | 1.00050 |

$$\boxed{GA = \frac{hl^2 E}{2\left[\dfrac{h^3}{A_c} + \dfrac{[(l/2)^2 + h^2]^{3/2}}{A_d}\right]}}$$

| | |
|---|---|
| GA = | 4198.30743 |
| EI = | 1470.00000 |
| alpha = | 1.68997 |

| x location | Deflection (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0027 |
| 66 | 0.0068 |
| 64 | 0.0114 |
| 62 | 0.0165 |
| 60 | 0.0220 |
| 58 | 0.0279 |
| 56 | 0.0342 |
| 54 | 0.0408 |
| 52 | 0.0478 |
| 50 | 0.0550 |
| 48 | 0.0624 |
| 46 | 0.0700 |
| 44 | 0.0779 |
| 42 | 0.0859 |
| 40 | 0.0941 |
| 38 | 0.1023 |
| 36 | 0.1107 |
| 34 | 0.1192 |
| 32 | 0.1277 |
| 30 | 0.1363 |
| 28 | 0.1449 |
| 26 | 0.1535 |
| 24 | 0.1621 |
| 22 | 0.1707 |
| 20 | 0.1792 |
| 18 | 0.1877 |
| 16 | 0.1962 |
| 14 | 0.2046 |
| 12 | 0.2129 |
| 10 | 0.2212 |
| 8 | 0.2294 |
| 6 | 0.2375 |
| 4 | 0.2455 |
| 2 | 0.2534 |
| 0 | 0.2612 |

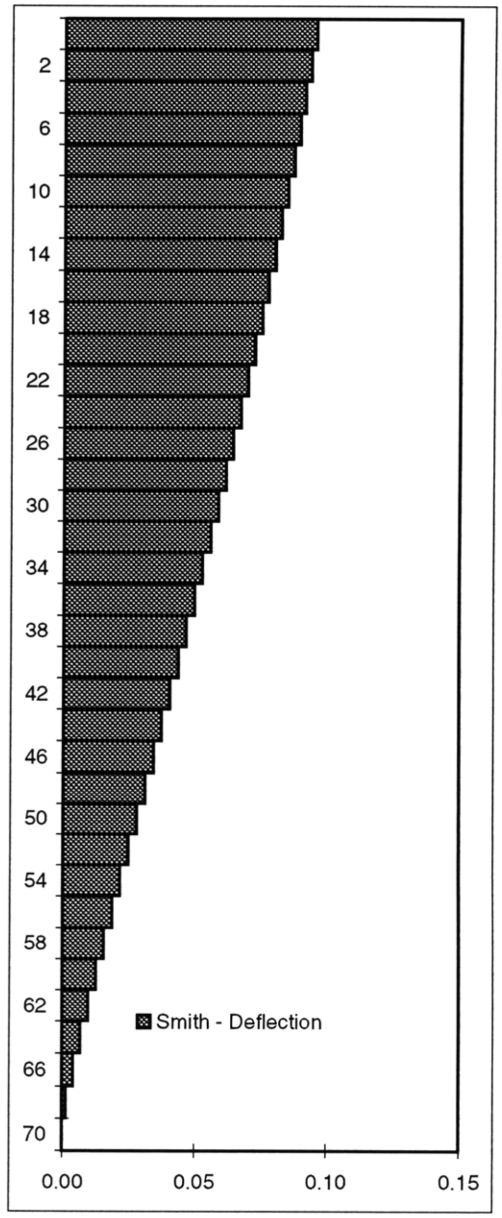


| | | |
|---|---|---|
| w | Uniform Load | 0.1 |
| H | Structure Height | 70 |
| h | Story Height | 10 |
| l | Frame Width | 10 |
| Ac | Area Col | 0.02816 |
| Ad | Area Dia | 0.00393 |
| E | Modulus | 4176000 |
| I | (Calculated) | 0.00141 |
| Ig | (Calculated) | 1.40945 |
| k | (Calculated) | 1.00050 |

$$GA = \frac{hl^2 E}{2\left[\dfrac{h^3}{A_c} + \dfrac{[(l/2)^2 + h^2]^{3/2}}{A_d}\right]}$$

| | |
|---|---|
| GA = | 5342.71317 |
| EI = | 5880.00000 |
| alpha = | 0.95322 |

| x location | Deflection (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0015 |
| 66 | 0.0041 |
| 64 | 0.0069 |
| 62 | 0.0097 |
| 60 | 0.0127 |
| 58 | 0.0156 |
| 56 | 0.0187 |
| 54 | 0.0217 |
| 52 | 0.0248 |
| 50 | 0.0279 |
| 48 | 0.0310 |
| 46 | 0.0341 |
| 44 | 0.0372 |
| 42 | 0.0403 |
| 40 | 0.0434 |
| 38 | 0.0465 |
| 36 | 0.0495 |
| 34 | 0.0525 |
| 32 | 0.0555 |
| 30 | 0.0584 |
| 28 | 0.0613 |
| 26 | 0.0641 |
| 24 | 0.0669 |
| 22 | 0.0697 |
| 20 | 0.0723 |
| 18 | 0.0750 |
| 16 | 0.0775 |
| 14 | 0.0800 |
| 12 | 0.0824 |
| 10 | 0.0847 |
| 8 | 0.0870 |
| 6 | 0.0892 |
| 4 | 0.0914 |
| 2 | 0.0934 |
| 0 | 0.0954 |



| | | |
|---|---|---|
| Wind Load | **b** | 0.1 |
| Height | **H** | 70 |
| Width | **B** | 10 |
| Bay Height | **l** | 10 |
| Dia Length | **L** | 14.14 |
| Dia Angle | **theta** | 45.00 |
| Elasticity | **E beam** | 4176000 |
| | **E col** | 4176000 |
| | **E diag** | 4176000 |
| Area | **A beam** | 0.01408 |
| | **A col** | 0.01408 |
| | **A diag** | 0.00393 |
| | $\boxed{f*}$ | 3 |

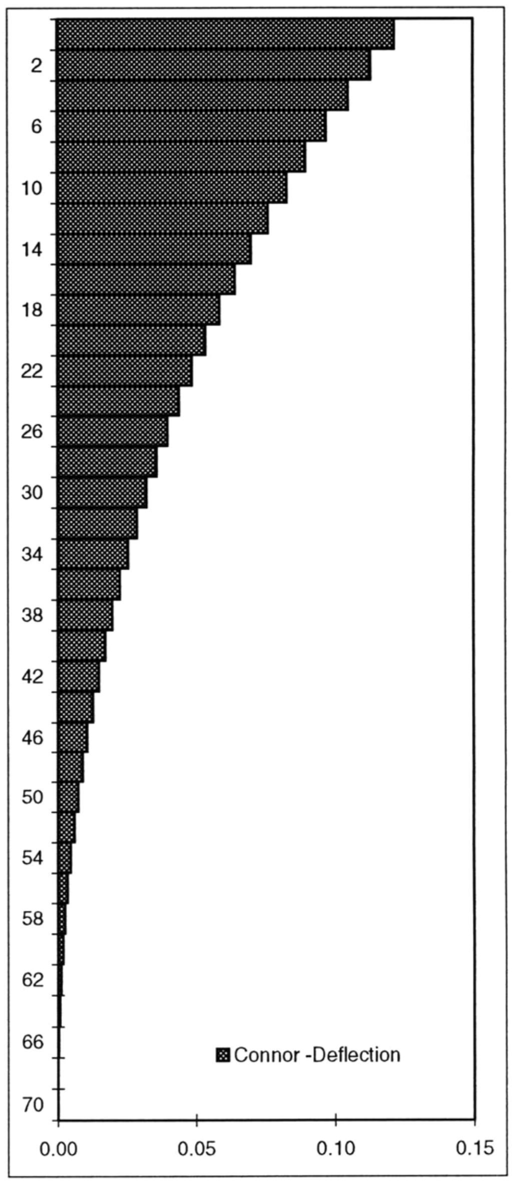
$$\boxed{Deflection = \frac{b(H-x)^4}{24D_B} + \frac{b(H-x)^2}{2D_T}}$$

$$\boxed{D_B = \frac{A^C E^C B^2}{2}}$$

$$\boxed{D_T = \frac{1}{\frac{2L^3}{lB^2A^DE^D} + \frac{l^2}{2B^2A^CE^C} + \frac{B}{4lA^BE^B}}}$$

| | | |
|---|---|---|
| Asp. Ratio | **B/H** | 1 to 7 |
| Calculate | **s** | 1.16667 |

| x Location | Defl (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0001 |
| 66 | 0.0003 |
| 64 | 0.0006 |
| 62 | 0.0011 |
| 60 | 0.0018 |
| 58 | 0.0026 |
| 56 | 0.0036 |
| 54 | 0.0047 |
| 52 | 0.0059 |
| 50 | 0.0074 |
| 48 | 0.0090 |
| 46 | 0.0108 |
| 44 | 0.0127 |
| 42 | 0.0149 |
| 40 | 0.0172 |
| 38 | 0.0198 |
| 36 | 0.0225 |
| 34 | 0.0255 |
| 32 | 0.0287 |
| 30 | 0.0322 |
| 28 | 0.0359 |
| 26 | 0.0399 |
| 24 | 0.0441 |
| 22 | 0.0487 |
| 20 | 0.0535 |
| 18 | 0.0586 |
| 16 | 0.0641 |
| 14 | 0.0699 |
| 12 | 0.0761 |
| 10 | 0.0827 |
| 8 | 0.0896 |
| 6 | 0.0969 |
| 4 | 0.1047 |
| 2 | 0.1129 |
| 0 | 0.1215 |



| | | |
|---|---|---|
| Wind Load | **b** | 0.1 |
| Height | **H** | 70 |
| Width | **B** | 10 |
| Bay Height | **l** | 10 |
| Dia Length | **L** | 14.14 |
| Dia Angle | **theta** | 45.00 |
| Elasticity | **E beam** | 4176000 |
| | **E col** | 4176000 |
| | **E diag** | 4176000 |
| Area | **A beam** | 0.01408 |
| | **A col** | 0.01408 |
| | **A diag** | 0.00197 |
| | $\boxed{f*}$ | 3 |

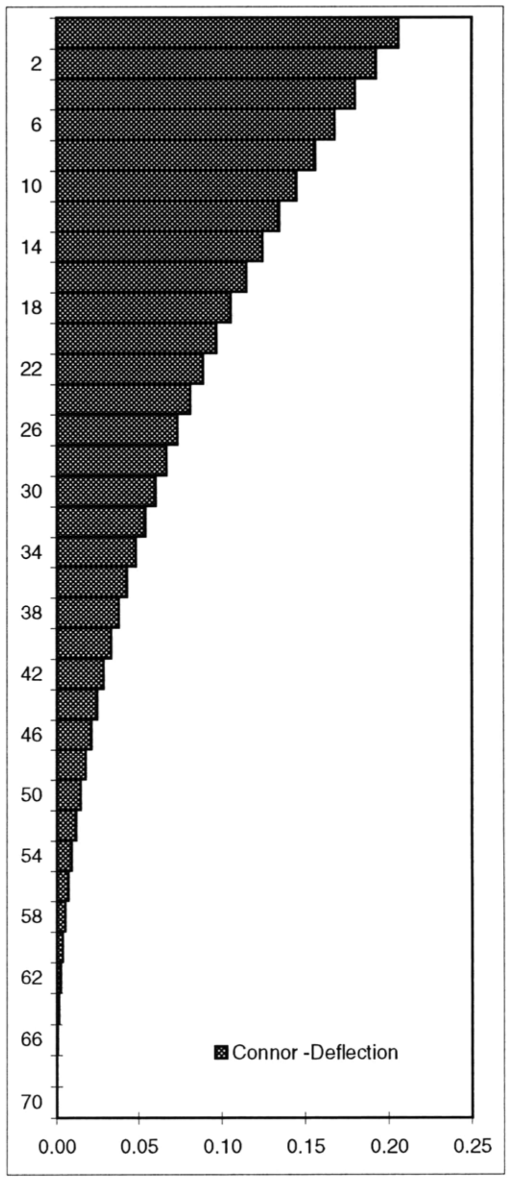
$$Deflection = \frac{b(H-x)^4}{24D_B} + \frac{b(H-x)^2}{2D_T}$$

$$D_B = \frac{A^C E^C B^2}{2}$$

$$D_T = \cfrac{1}{\cfrac{2L^3}{lB^2A^DE^D} + \cfrac{l^2}{2B^2A^CE^C} + \cfrac{B}{4lA^BE^B}}$$

| | | |
|---|---|---|
| Asp. Ratio | **B/H** | 1 to 7 |
| Calculate | **s** | 1.16667 |

| x Location | Defl (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0001 |
| 66 | 0.0006 |
| 64 | 0.0013 |
| 62 | 0.0023 |
| 60 | 0.0035 |
| 58 | 0.0051 |
| 56 | 0.0069 |
| 54 | 0.0091 |
| 52 | 0.0115 |
| 50 | 0.0143 |
| 48 | 0.0173 |
| 46 | 0.0207 |
| 44 | 0.0244 |
| 42 | 0.0284 |
| 40 | 0.0327 |
| 38 | 0.0374 |
| 36 | 0.0424 |
| 34 | 0.0478 |
| 32 | 0.0536 |
| 30 | 0.0598 |
| 28 | 0.0663 |
| 26 | 0.0732 |
| 24 | 0.0806 |
| 22 | 0.0883 |
| 20 | 0.0965 |
| 18 | 0.1052 |
| 16 | 0.1143 |
| 14 | 0.1239 |
| 12 | 0.1340 |
| 10 | 0.1446 |
| 8 | 0.1558 |
| 6 | 0.1675 |
| 4 | 0.1797 |
| 2 | 0.1925 |
| 0 | 0.2059 |



| | | |
|---|---|---|
| Wind Load | **b** | 0.1 |
| Height | **H** | 70 |
| Width | **B** | 10 |
| Bay Height | **l** | 10 |
| Dia Length | **L** | 14.14 |
| Dia Angle | **theta** | 45.00 |
| Elasticity | **E beam** | 4176000 |
| | **E col** | 4176000 |
| | **E diag** | 4176000 |
| Area | **A beam** | 0.01408 |
| | **A col** | 0.01408 |
| | **A diag** | 0.00787 |
| | $\boxed{f*}$ | 3 |

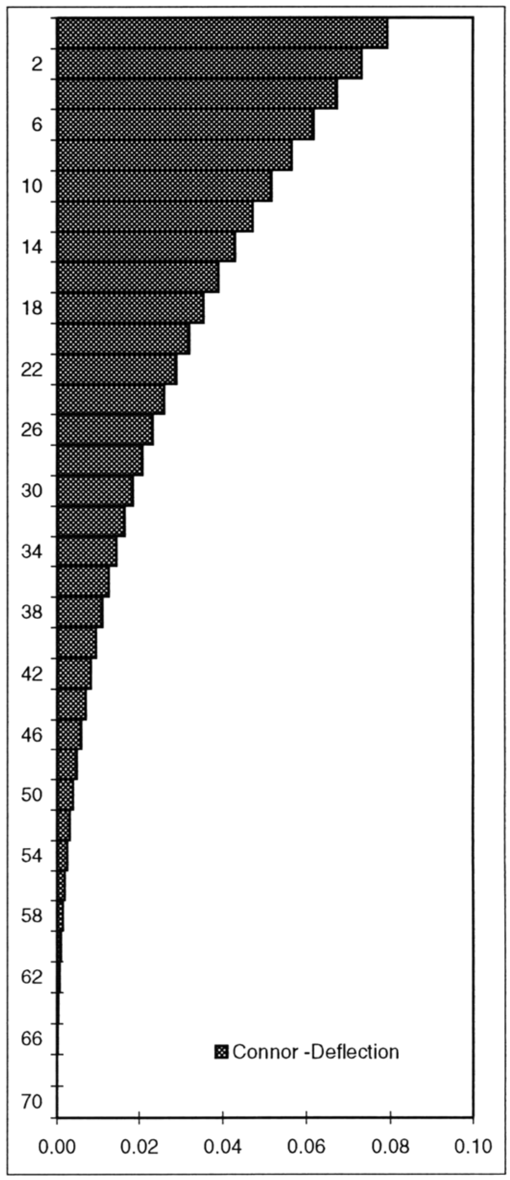
$$\boxed{Deflection = \frac{b(H-x)^4}{24D_B} + \frac{b(H-x)^2}{2D_T}}$$

$$\boxed{D_B = \frac{A^C E^C B^2}{2}}$$

$$\boxed{D_T = \frac{1}{\dfrac{2L^3}{lB^2A^DE^D} + \dfrac{l^2}{2B^2A^CE^C} + \dfrac{B}{4lA^BE^B}}}$$

| | | |
|---|---|---|
| Asp. Ratio | B/H | 1 to 7 |
| Calculate | **s** | 1.16667 |

| x Location | Defl (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0000 |
| 66 | 0.0001 |
| 64 | 0.0003 |
| 62 | 0.0006 |
| 60 | 0.0009 |
| 58 | 0.0014 |
| 56 | 0.0019 |
| 54 | 0.0025 |
| 52 | 0.0031 |
| 50 | 0.0039 |
| 48 | 0.0048 |
| 46 | 0.0058 |
| 44 | 0.0069 |
| 42 | 0.0081 |
| 40 | 0.0095 |
| 38 | 0.0110 |
| 36 | 0.0126 |
| 34 | 0.0144 |
| 32 | 0.0163 |
| 30 | 0.0184 |
| 28 | 0.0207 |
| 26 | 0.0232 |
| 24 | 0.0259 |
| 22 | 0.0288 |
| 20 | 0.0320 |
| 18 | 0.0354 |
| 16 | 0.0390 |
| 14 | 0.0429 |
| 12 | 0.0471 |
| 10 | 0.0517 |
| 8 | 0.0565 |
| 6 | 0.0617 |
| 4 | 0.0672 |
| 2 | 0.0731 |
| 0 | 0.0793 |



| | | |
|---|---|---|
| Wind Load | **b** | 0.1 |
| Height | **H** | 70 |
| Width | **B** | 10 |
| Bay Height | **l** | 10 |
| Dia Length | **L** | 14.14 |
| Dia Angle | **theta** | 45.00 |
| Elasticity | **E beam** | 4176000 |
| | **E col** | 4176000 |
| | **E diag** | 4176000 |
| Area | **A beam** | 0.00704 |
| | **A col** | 0.00704 |
| | **A diag** | 0.00393 |
| | $\boxed{f*}$ | 3 |

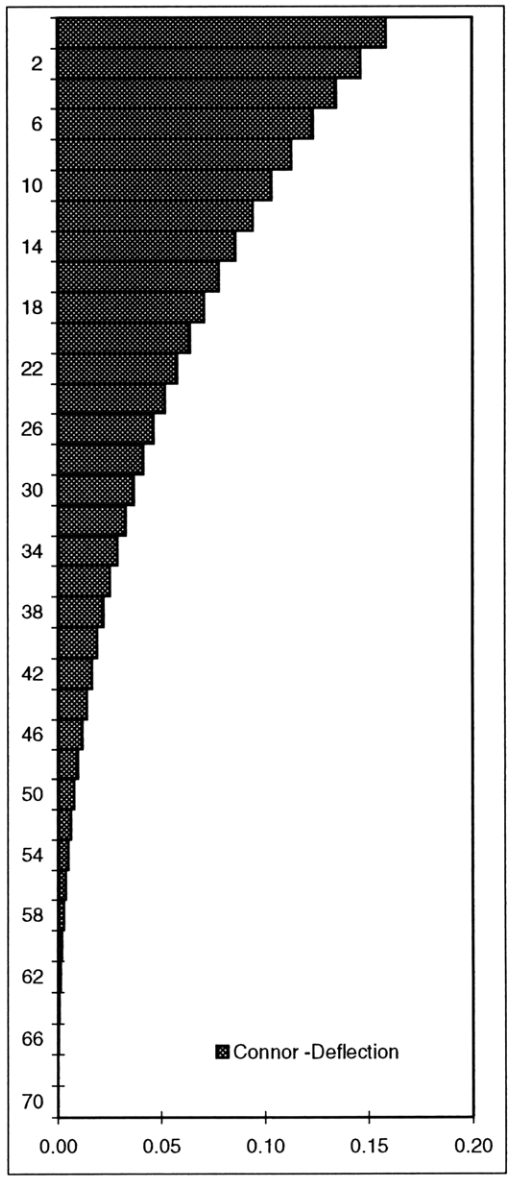
$$\boxed{Deflection = \frac{b(H-x)^4}{24D_B} + \frac{b(H-x)^2}{2D_T}}$$

$$\boxed{D_B = \frac{A^C E^C B^2}{2}}$$

$$\boxed{D_T = \frac{1}{\frac{2L^3}{lB^2A^DE^D} + \frac{l^2}{2B^2A^CE^C} + \frac{B}{4lA^BE^B}}}$$

| | | |
|---|---|---|
| Asp. Ratio | B/H | 1 to 7 |
| Calculate | s | 1.16667 |

| x Location | Defl (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0001 |
| 66 | 0.0003 |
| 64 | 0.0007 |
| 62 | 0.0012 |
| 60 | 0.0019 |
| 58 | 0.0027 |
| 56 | 0.0037 |
| 54 | 0.0049 |
| 52 | 0.0063 |
| 50 | 0.0079 |
| 48 | 0.0096 |
| 46 | 0.0116 |
| 44 | 0.0138 |
| 42 | 0.0162 |
| 40 | 0.0189 |
| 38 | 0.0219 |
| 36 | 0.0252 |
| 34 | 0.0287 |
| 32 | 0.0326 |
| 30 | 0.0368 |
| 28 | 0.0414 |
| 26 | 0.0464 |
| 24 | 0.0518 |
| 22 | 0.0577 |
| 20 | 0.0640 |
| 18 | 0.0707 |
| 16 | 0.0780 |
| 14 | 0.0859 |
| 12 | 0.0943 |
| 10 | 0.1033 |
| 8 | 0.1130 |
| 6 | 0.1233 |
| 4 | 0.1343 |
| 2 | 0.1461 |
| 0 | 0.1587 |



| | | |
|---|---|---|
| Wind Load | **b** | 0.1 |
| Height | **H** | 70 |
| Width | **B** | 10 |
| Bay Height | **l** | 10 |
| Dia Length | **L** | 14.14 |
| Dia Angle | **theta** | 45.00 |
| Elasticity | **E beam** | 4176000 |
| | **E col** | 4176000 |
| | **E diag** | 4176000 |
| Area | **A beam** | 0.02816 |
| | **A col** | 0.02816 |
| | **A diag** | 0.00393 |
| | $\boxed{f*}$ | 3 |

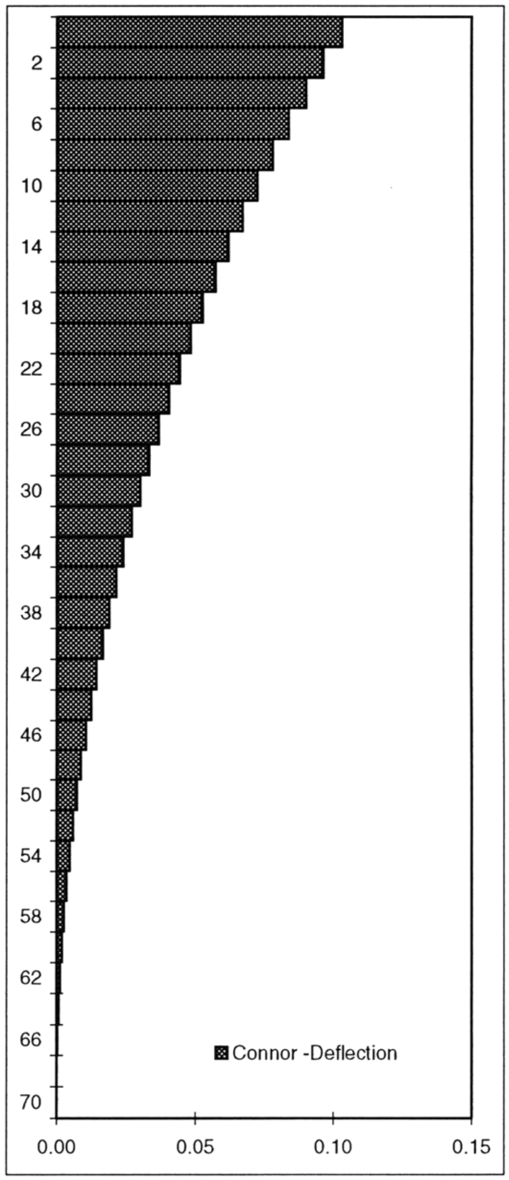
$$\boxed{Deflection = \frac{b(H-x)^4}{24 D_B} + \frac{b(H-x)^2}{2 D_T}}$$

$$\boxed{D_B = \frac{A^C E^C B^2}{2}}$$

$$\boxed{D_T = \frac{1}{\frac{2L^3}{lB^2 A^D E^D} + \frac{l^2}{2B^2 A^C E^C} + \frac{B}{4lA^B E^B}}}$$

| | | |
|---|---|---|
| Asp. Ratio | B/H | 1 to 7 |
| Calculate | s | 1.16667 |

| x Location | Defl (ft) |
|---|---|
| 70 | 0.0000 |
| 68 | 0.0001 |
| 66 | 0.0003 |
| 64 | 0.0006 |
| 62 | 0.0011 |
| 60 | 0.0018 |
| 58 | 0.0025 |
| 56 | 0.0035 |
| 54 | 0.0045 |
| 52 | 0.0058 |
| 50 | 0.0071 |
| 48 | 0.0087 |
| 46 | 0.0103 |
| 44 | 0.0122 |
| 42 | 0.0142 |
| 40 | 0.0164 |
| 38 | 0.0187 |
| 36 | 0.0212 |
| 34 | 0.0239 |
| 32 | 0.0268 |
| 30 | 0.0299 |
| 28 | 0.0331 |
| 26 | 0.0366 |
| 24 | 0.0403 |
| 22 | 0.0442 |
| 20 | 0.0483 |
| 18 | 0.0526 |
| 16 | 0.0572 |
| 14 | 0.0620 |
| 12 | 0.0670 |
| 10 | 0.0723 |
| 8 | 0.0779 |
| 6 | 0.0837 |
| 4 | 0.0898 |
| 2 | 0.0962 |
| 0 | 0.1030 |



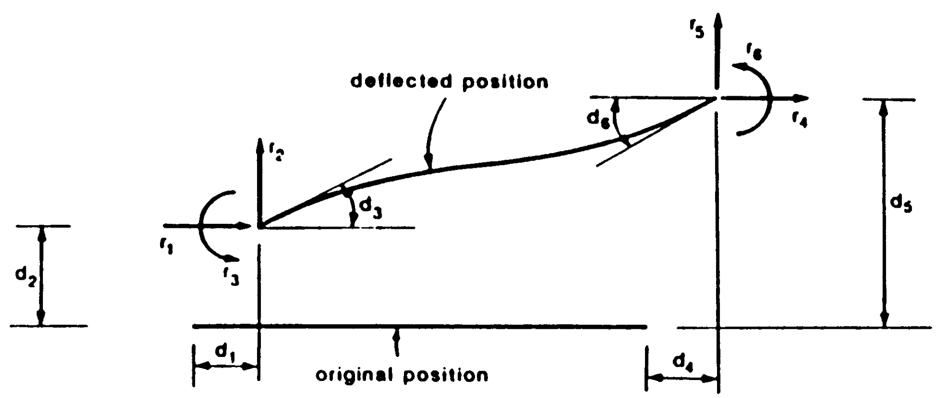
## *Appendix V*

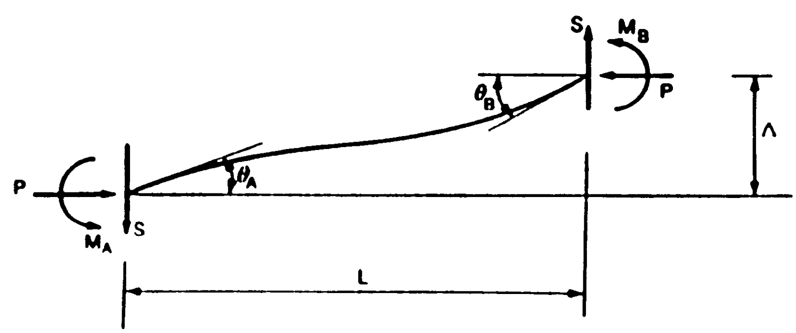
# BEAM - COLUMN MATRIX FORMULATION

*Page*

# Definitions

## Beam – Column Stiffness Matrix Formulation



## 6x6 Structural Stiffness Matrix Formulation

$$\begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{pmatrix} = \begin{bmatrix} & & \\ & K & \\ & & \end{bmatrix} \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{pmatrix}$$

$$\begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{pmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{pmatrix} M_A \\ M_B \\ S \\ P \end{pmatrix}$$

$$\begin{pmatrix} M_A \\ M_B \\ S \\ P \end{pmatrix} = \begin{bmatrix} S_{ii} & S_{ij} & -\frac{sii+sij}{L} & 0 \\ S_{ij} & S_{ii} & -\frac{sii+sij}{L} & 0 \\ -\frac{sii+sij}{L} & -\frac{sii+sij}{L} & \frac{2(sii+sij)-(kL)^2}{L^2} & 0 \\ 0 & 0 & 0 & \frac{A}{I} \end{bmatrix} \begin{pmatrix} \theta_A \\ \theta_B \\ \Delta \\ u \end{pmatrix}$$

$$\begin{pmatrix} \theta_A \\ \theta_B \\ \Delta \\ u \end{pmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{pmatrix}$$

$$\begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{pmatrix} = \begin{bmatrix} \frac{A}{I} & 0 & 0 & -\frac{A}{I} & 0 & 0 \\ 0 & \frac{12}{L^2}\phi_1 & \frac{6}{L}\phi_2 & 0 & -\frac{12}{L^2}\phi_1 & \frac{6}{L}\phi_2 \\ 0 & \frac{6}{L}\phi_2 & 4\phi_3 & 0 & -\frac{6}{L}\phi_2 & 2\phi_4 \\ -\frac{A}{I} & 0 & 0 & \frac{A}{I} & 0 & 0 \\ 0 & -\frac{12}{L^2}\phi_1 & -\frac{6}{L}\phi_2 & 0 & \frac{12}{L^2}\phi_1 & -\frac{6}{L}\phi_2 \\ 0 & \frac{6}{L}\phi_2 & 2\phi_4 & 0 & -\frac{6}{L}\phi_2 & 4\phi_3 \end{bmatrix} \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{pmatrix}$$

## 4x4 Warping Torsion Matrix Formulation

$$\boxed{\begin{aligned} Sv &= \sinh(kvL) \\ Cv &= \cosh(kvL) \\ kv &= \sqrt{\frac{GJ + \bar{k}}{EI_w}} \end{aligned}}$$

kvL = effective length for torsional buckling

$$\begin{bmatrix} M_{VA} \\ M_{WA}kv \\ M_{VB} \\ M_{VB}kv \end{bmatrix} = [Stiffness] \begin{bmatrix} \theta_{VA}kv \\ W_A \\ \theta_{VB}kv \\ W_B \end{bmatrix}$$

$$[Stiffness] = \frac{kv^2 EI_w}{2 - 2Cv + kvLSv} \begin{bmatrix} Sv & Cv - 1 & -Sv & Cv - 1 \\ Cv - 1 & kvLCv - Sv & 1 - Cv & Sv - kvL \\ -Sv & 1 - Cv & Sv & 1 - Cv \\ Cv - 1 & Sv - kvL & 1 - Cv & kvLCv - Sv \end{bmatrix}$$

*Appendix VII*

## BEAM - COLUMN MATLAB CODE

# Main File    Beam - Column Matlab Code

```
%********************************************************************************
% Gather information from a data file. Length and Angle of each member is computed
%********************************************************************************

% clear and input numbers from data.m file
clear
home
filename = input('Enter the name of the file:');
in_file = fopen(filename, 'r');
% reading from the data file

anything = fscanf( in_file, '%*s %*s %s',1);
no_nodes = fscanf( in_file, '%d', 1 );
anything = fscanf( in_file, '%*s %*s %s',1);
no_members = fscanf(in_file, '%d', 1 );
anything = fscanf( in_file, '%s',1);

% Input node data         1-nn 2-x 3-y 4-Ax 5-Sh 6-Mo 7-To 8-Bi 9-R1 10-R2
                          11-R3 12-R4 13-R5 14-D1 15-D2 16-D3 17-D4 18-D5
                          19-Ks1 20-Ks2 21-Ks3 22-Ks4 23-Ks5

anything = fscanf( in_file, '%*s %*s %*s %*s %*s %*s %*s %*s %*s %*s %*s %*s %*s %*s %*s %*s
%*s %*s %*s %*s %*s %*s %s',1);
[node_data , count] = fscanf( in_file, '%d %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f
%f %f %d', [23,no_nodes]);
count;
anything = fscanf( in_file, '%s',1);

% Input member data       1-mn 2-n+ 3-n- 4-A 5-E 6-I 7-k 8-Iw 9-kv 10-PF 11-a
                          12-b 13-w 14-lin 15-grT 16-dtT

anything = fscanf( in_file, '%*s %*s %*s %*s %*s %*s %*s %*s %*s %*s %*s %*s %*s %*s %*s %s',1);
member_data = fscanf( in_file, '%d %d %f %f %f %f %f %f %f %f %f %f %f %f %f %d', [16,
no_members]);

fclose(in_file);
node_data_view = node_data';
member_data_view = member_data';


DOF = no_nodes * 5

t = [    0     0     1     0     0     0
         0     0     0     0     0     1
         0    -1     0     0     1     0
         1     0     0    -1     0     0];       % transpose matrix

Kg = zeros(DOF, DOF);
FGSIX_DISP = zeros(DOF,1);
FG_TEMP = zeros(DOF,1);
FG_FEF = zeros(DOF,1);
```

```
% mn = member number
mn = 1;
while mn <= no_members

        home
        member = mn
        prompt = input ('Press enter to see properties');                    %%%%%
        % Gather member data;

        A = member_data(4,mn)
        E = member_data(5,mn)
        I = member_data(6,mn)
%       k = member_data(7,mn)
        Iw = member_data(8,mn)
        kv = member_data(9,mn)



        prompt = input ('Press enter to see Angle and Length');              %%%%%
        % Compute Angle and Length

        pos_node = member_data(2,member);                 % Pos Node
        pos_x = node_data(2,pos_node);
        pos_y = node_data(3,pos_node);

        neg_node = member_data(3,member);                 % Neg Node
        neg_x = node_data(2,neg_node);
        neg_y = node_data(3,neg_node);

        delta_x = pos_x - neg_x;
        delta_y = pos_y - neg_y;

        l = sqrt(delta_x^2 + delta_y^2)             % Length

        Ang = atan(delta_y / delta_x)               % Angle

%       kl = k*l;

        %**************************************************************************
        % Direction Matrix
        %**************************************************************************

        prompt = input ('Press enter to compute direction matrix');          %%%%%
        A = Ang
        C = [   cos(A)  sin(A)  0       0       0       0       0       0       0       0
                -sin(A) cos(A)  0       0       0       0       0       0       0       0
                0       0       1       0       0       0       0       0       0       0
                0       0       0       cos(A)  sin(A)  0       0       0       0       0
                0       0       0       -sin(A) cos(A)  0       0       0       0       0
                0       0       0       0       0       cos(A)  sin(A)  0       0       0
                0       0       0       0       0       -sin(A) cos(A)  0       0       0
                0       0       0       0       0       0       0       1       0       0
                0       0       0       0       0       0       0       0       cos(A)  sin(A)
                0       0       0       0       0       0       0       0       -sin(A) cos(A)]
```

```
Csix = zeros(6,6);
Csix(1:3,1:3) = C(1:3,1:3);
Csix(4:6,4:6) = C(6:8,6:8);


%**************************************************************************
% In order to construct the global K matrix - the program will first assemble the
% elemental stiffness matrix for each member. The first step in this procedure
% is to determine whether the axial force in each member is compressive or
% tensile. In this determination, the program will assemble the force matrix
% for each member (summation of nodal forces, temperature, fixed end forces,
% and support displacement). The global force matrix is also assembled during
% this process. After the final assembly of each member force vector, the program
% breaks down the 6x1 force matrix into a 4x1 matrix with axial force, shear force,
% moment at A, and moment at B as its components. The axial force can then be
% be determined.
%**************************************************************************

prompt = input ('Press enter to compute member and global force matrix');          %%%%%


% Assume no axial force
        sii = 4;
        sij = 2;
        kl = 0;                    % k = 0 since P = 0          Forming trial kstarsixt

        f_ksix

%**************************************************************************
% FMSIX_NF = Forces on Frame Nodes
%**************************************************************************

FMSIX_NF = zeros(6,1);
        FMSIX_NF(1:3,1) = node_data(4:6,neg_node);
        FMSIX_NF(4:6,1) = node_data(4:6,pos_node);
        FMSIX_NF = Csix * FMSIX_NF;


%**************************************************************************
% FMSIX_DISP = Forces due to Support Displacement
%**************************************************************************

FMSIX_DISP = zeros(6,1);
        UM_DISP(1:3,1) = node_data(14:16,neg_node);
        UM_DISP(4:6,1) = node_data(14:16,pos_node);


if UM_DISP == zeros(6,1)
else
        f_KeArr
end
```

```
%***************************************************************************
% FM_TEMP = Forces due to Temperature Changes
%***************************************************************************

FM_TEMP = zeros(6,1);
            gradT = member_data(15,member);
            deltaT = member_data(16,member);

if deltaT == 0
else
          f_TEMP
end


%***************************************************************************
% FMSIX_FEF = Fixed End Forces
%***************************************************************************

FMSIX_FEF = zeros(6,1);
PF = member_data(10,mn);
w = member_data(13,mn);
lin = member_data(14,mn);

if PF + w + lin == 0
else
          f_FEF
end


%***************************************************************************
% FM and FG = Member and Global Force Vectors
%***************************************************************************

prompt = input ('Press enter to see FM');                              %%%%%%
FM = FMSIX_NF + FMSIX_DISP + FM_TEMPT + FMSIX_FEF

FG_DTF = FGSIX_DISP + FG_TEMP + FG_FEF;

prompt = input ('Press enter to see rstar');                           %%%%%%
rstar = t*FM


%***************************************************************************
% Check to see if P is Compressive or Tensile
%***************************************************************************

P = rstar(4,1);

if P <= .05 & P>=-.05
          P = 0;
end
```

```
%*************************************************************************
% Define Terms
%*************************************************************************

k = sqrt(P/(E*I));
kl = k*l;

c = 4;
s = 2;

if P == 0 else
          siic = (kl * sin(kl) - (kl^2) * cos(kl)) / (2 - 2*cos(kl) - kl*sin(kl));
          sijc = (kl^2 - kl * sin(kl)) / (2 - 2 * cos(kl) - kl*sin(kl));
          siit = (kl^2 * cosh(kl) - kl * sinh(kl)) / (2 - 2*cosh(kl) - kl*sinh(kl));
          sijt = (kl * sinh(kl) - kl^2) / (2 - 2*cosh(kl) - kl*sinh(kl));
end


          if P < 0
                    p = input ('Axial force is Negative');
                    p = 1;
                    sii = siit;
                    sij = sijt;
          end

          if P > 0
                    p = input ('Axial force is Positive');
                    p = 2;
                    sii = siic;
                    sij = sijc;
          end

          if P == 0
                    p = input ('There is no axial force');
                    p = 3;
                    sii = c;
                    sij = s;
          end


%*************************************************************************
% Construct kstarsix with Known P Direction
%*************************************************************************

prompt = input ('Press enter to form kstarsix');                    %%%%%%

f_ksix

ksix = kstarsix                                                % Final ksix
```

```
%*****************************************************************************
% Form kfour - Lateral Torsion and Warping
%*****************************************************************************

kfour = zeros(4,4)
if Iw == 0
else
        f_LTB
end


%*****************************************************************************
% kten Assembly
%*****************************************************************************

prompt = input ('Press enter to compute kten');
%%%%%

kten = zeros(10,10);
prompt = input ('Press enter key to see K 10x10 matrix');
kten(1:3,1:3) = ksix(1:3,1:3);
kten(6:8,6:8) = ksix(4:6,4:6);
kten(4:5,4:5) = kfour(1:2,1:2);
kten(9:10,9:10) = kfour(3:4,3:4);
kten                                                    % Final kten


%*****************************************************************************
% Elemental K Matrix (Kel) and Global Matrix
%*****************************************************************************

prompt = input ('Press enter to compute Kel');
%%%%%
Kel = C'*kten*C

prompt = input ('Press enter to insert direction matrix into Kg');       %%%%%
negn = neg_node;
posn = pos_node;

Kg(5*posn-4:5*posn, 5*posn-4:5*posn) = Kg(5*posn-4:5*posn,
        5*posn-4:5*posn)+Kel(1:5,1:5);
Kg(5*negn-4:5*negn, 5*negn-4:5*negn) = Kg(5*negn-4:5*negn,
        5*negn-4:5*negn)+Kel(6:10,6:10);
Kg(5*negn-4:5*negn, 5*posn-4:5*posn) = Kg(5*negn-4:5*negn,
        5*posn-4:5*posn)-Kel(1:5,6:10);
Kg(5*posn-4:5*posn, 5*negn-4:5*negn) = Kg(5*posn-4:5*posn,
        5*negn-4:5*negn)-Kel(6:10,1:5);
Kg


prompt = input ('Press enter to continue');                              %%%%%

mn = mn + 10;
end
```

```
%****************************************************************************
% Input Springs into Kg
%****************************************************************************

FG_NF = zeros(DOF,1);
nn = 1;
while nn <= no_nodes

	%****************************************************************************
	% Add in K spring into Kg
	%****************************************************************************

	Kg(5*nn-4,5*nn-4) = Kg(5*nn-4,5*nn-4) + node_data(19,nn);
	Kg(5*nn-3,5*nn-3) = Kg(5*nn-3,5*nn-3) + node_data(20,nn);
	Kg(5*nn-2,5*nn-2) = Kg(5*nn-2,5*nn-2) + node_data(21,nn);
	Kg(5*nn-1,5*nn-1) = Kg(5*nn-1,5*nn-1) + node_data(22,nn);
	Kg(5*nn  ,5*nn  ) = Kg(5*nn  ,5*nn  ) + node_data(23,nn);


	%****************************************************************************
	% Add up Nodal Forces
	%****************************************************************************

	FG_NF(5*nn-4:5*nn,1) = node_data(4:8,nn);

	nn = nn + 1;
end


%****************************************************************************
% Add up Global Force Vector - FG_NF + FG_DTF
%****************************************************************************

FG = FG_NF + FG_DTF


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%****************************************************************************
% Rearrange all matrices to take into account those joints that are fixed vs free
%****************************************************************************

num_free = 0
nn = 1;
while nn <= no_nodes
	ar = node_data(9,nn);		% axial restraint
	sr = node_data(10,nn);		% shear restraint
	mr = node_data(11,nn);		% moment restraint
	tr = node_data(12,nn);		% torsion restraint
	br = node_data(13,nn);		% bi-moment restraint
```

```
        if ar == 1; num_free = num_free + 1; end
        if sr == 1; num_free = num_free + 1; end
        if mr == 1; num_free = num_free + 1; end
        if tr == 1; num_free = num_free + 1; end
        if br == 1; num_free = num_free + 1; end

        nn = nn + 1
end

%*******************************************************************************
% Arrange into new vectors - Kgr, FGr, Ur
%*******************************************************************************

        f_KgArr

Kgr_set = Kgr(1:num_free, 1:num_free);
FGr_set = FGr(1:num_free,1);
Disp_set = Kgr_set \ FGr_set;


%*******************************************************************************
% Put Disp_set back into original order
%*******************************************************************************

Disp = zeros(DOF,1)
nf = 1;
while nf <= num_free
        Disp(Ur(nf)) = Disp_set(nf);
        nf = nf + 1
end


%*******************************************************************************
% Member Displacements
%*******************************************************************************

mn = 1;
MEA = zeros(no_members,1);
while mn <= no_members
         member = mn
         pos_node = member_data(2,member);
         neg_node = member_data(3,member);
        da = Disp(5*pos_node-4,1) - Disp(5*neg_node-4,1);
        ds = Disp(5*pos_node-3,1) - Disp(5*neg_node-3,1);
        dm = Disp(5*pos_node-2,1) - Disp(5*neg_node-2,1);
        dt = Disp(5*pos_node-1,1) - Disp(5*neg_node-1,1);
        db = Disp(5*pos_node  ,1) - Disp(5*neg_node  ,1);

        mn = mn + 1
end
```

# 6 x 6 Stiffness Matrix Formulation

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% f_ksix.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


kstarsix = zeros(4,4);

kstarsix(1,1) = sii;
kstarsix(2,2) = sii;
kstarsix(1,2) = sij;
kstarsix(2,1) = sij;
kstarsix(1,3) = (sii + sij)/(-l);
kstarsix(3,1) = (sii + sij)/(-l);
kstarsix(2,3) = (sii + sij)/(-l);
kstarsix(3,2) = (sii + sij)/(-l);
kstarsix(1,4) = 0;
kstarsix(2,4) = 0;
kstarsix(3,4) = 0;
kstarsix(4,1) = 0;
kstarsix(4,2) = 0;
kstarsix(4,3) = 0;
kstarsix(3,3) = ((sii + sij) - kl^2) / l^2;
kstarsix(4,4) = A/l;

kstarsix = (E*I/l) * kstarsix
kstarsix = t'*kstarsix*t;
```

## 4 x 4 Warping and Torsion Matrix Formulation

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% f_LTB.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

kvl = kv * l;

Sv = sinh(kvl);
Cv = cosh(kvl);
kcoef = (kv^2*E*Iw) / (2 - 2*Cv + kvl*Sv);
kstarfour = zeros(4,4);                          % Forming kstarfour
kstarfour(1,1) = Sv;
kstarfour(3,3) = Sv;

kstarfour(1,4) = Cv - 1;
kstarfour(4,1) = Cv - 1;
kstarfour(1,2) = Cv - 1;
kstarfour(2,1) = Cv - 1;

kstarfour(1,3) = -Sv;
kstarfour(3,1) = -Sv;

kstarfour(3,4) = -Cv + 1;
kstarfour(3,2) = -Cv + 1;
kstarfour(4,3) = -Cv + 1;
kstarfour(2,3) = -Cv + 1;

kstarfour(4,2) = Sv - kvl;
kstarfour(2,4) = Sv - kvl;

kstarfour(2,2) = kvl * Cv - Sv;
kstarfour(4,4) = kvl * Cv - Sv;

prompt = input ('Press enter key to see K 4x4 matrix');          %%%%%
kfour = kcoef * kstarfour;
kfour                                            % Final kfour
```

## Fixed End Forces

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% f_FEF.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%-------------------------------------------------------------------------------
% FEF_P = Fixed End Forces due to Point Loads
%-------------------------------------------------------------------------------
   FEF_P = zeros(6,1);
        a = member_data(11,mn);
        b = member_data(12,mn);

         if PF == 0;
                a = l/2;
                b = l/2;
         end

         FEF_P(1,1) = 0;
         FEF_P(4,1) = 0;
         FEF_P(2,1) = ((PF*b^2)*(3*a + b) / l^3);
         FEF_P(5,1) = ((PF*a^2)*(3*b + a) / l^3);
         FEF_P(3,1) = (PF*a*b^2) / l^2;
         FEF_P(6,1) = (PF*a^2*b) / l^2;

%-------------------------------------------------------------------------------
% FEF_w = Fixed End Forces due to Uniform Loads
%-------------------------------------------------------------------------------
   FEF_w = zeros(6,1);
         FEF_w(1,1) = 0;
         FEF_w(4,1) = 0;
         FEF_w(2,1) = (w*l) / 2;
         FEF_w(5,1) = (w*l) / 2;
         FEF_w(3,1) = (w*l^2) / 12;
         FEF_w(6,1) = (w*l^2) / 12;

%-------------------------------------------------------------------------------
% FEF_lin = Fixed End Forces due to Linearly Varying Loads
%-------------------------------------------------------------------------------
   FEF_lin = zeros(6,1);    % non zero load is at the positive end
         FEF_lin(1,1) = 0;
         FEF_lin(4,1) = 0;
         FEF_lin(2,1) = (3*lin*l) / 20;
         FEF_lin(5,1) = (7*lin*l) / 20;
         FEF_lin(3,1) = (lin*l^2) / 30;
         FEF_lin(6,1) = (lin*l^2) / 20;
```

```
%		FEF_P
%		FEF_w
%		FEF_lin

FMSIX_FEF = FEF_P + FEF_w + FEF_lin;
FMSIX_FEFT = Csix * FMSIX_FEF;

FG_FEF(5*neg_node-4:5*neg_node-2) = FG_FEF(5*neg_node-4:5*neg_node-2) +
FMSIX_FEFT(1:3,1);
FG_FEF(5*pos_node-4:5*pos_node-2) = FG_FEF(5*pos_node-4:5*pos_node-2) +
FMSIX_FEFT(4:6,1);
```

## Forces due to Displacements

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% f_KeArr.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


Num_freeM = 0

nn = neg_node
ad = node_data(9,nn)
sd = node_data(10,nn)
md = node_data(11,nn)
if ad == 0; Num_freeM = Num_freeM + 1; end
if sd == 0; Num_freeM = Num_freeM + 1; end
if md == 0; Num_freeM = Num_freeM + 1; end
nn = pos_node
ad = node_data(9,nn)
sd = node_data(10,nn)
md = node_data(11,nn)
if ad == 0; Num_freeM = Num_freeM + 1; end
if sd == 0; Num_freeM = Num_freeM + 1; end
if md == 0; Num_freeM = Num_freeM + 1; end

KgrrM = zeros(6,6)
placefreeM = 1
placefixedM = Num_freeM + 1
nn = neg_node
ad = node_data(9,nn)
sd = node_data(10,nn)
md = node_data(11,nn)

if ad == 1
        KgrrM(placefreeM,1:DOF) = Kstarsixt(1,1:DOF);
        DispM(placefreeM,1) = UM_DISP(1,1);
        FrM(placefreeM,1) = 1;
        placefreeM = placefreeM + 1;
else
        KgrrM(placefixedM,1:DOF) = Kstarsixt(1,1:DOF);
        DispM(placefixedM,1) = UM_DISP(1,1);
        FrM(placefixedM,1) = 1;
        placefixedM = placefixedM + 1;
end

if sd == 1
        KgrrM(placefreeM,1:DOF) = Kstarsixt(2,1:DOF);
        DispM(placefreeM,1) = UM_DISP(2,1);
        FrM(placefreeM,1) = 2;
        placefreeM = placefreeM + 1;
else
        KgrrM(placefixedM,1:DOF) = Kstarsixt(2,1:DOF);
        DispM(placefixedM,1) = UM_DISP(2,1);
```

```
		FrM(placefixedM,1) = 2;
		placefixedM = placefixedM + 1;
end

if md == 1
		KgrrM(placefreeM,1:DOF) = Kstarsixt(3,1:DOF);
		DispM(placefreeM,1) = UM_DISP(3,1);
		FrM(placefreeM,1) = 3;
		placefreeM = placefreeM + 1;
else
		KgrrM(placefixedM,1:DOF) = Kstarsixt(3,1:DOF);
		DispM(placefixedM,1) = UM_DISP(3,1);
		FrM(placefixedM,1) = 3;
		placefixedM = placefixedM + 1;
end

nn = pos_node
ad = node_data(9,nn)
sd = node_data(10,nn)
md = node_data(11,nn)

if ad == 1
		KgrrM(placefreeM,1:DOF) = Kstarsixt(4,1:DOF);
		DispM(placefreeM,1) = UM_DISP(4,1);
		FrM(placefreeM,1) = 4;
		placefreeM = placefreeM + 1;
else
		KgrrM(placefixedM,1:DOF) = Kstarsixt(4,1:DOF);
		DispM(placefixedM,1) = UM_DISP(4,1);
		FrM(placefixedM,1) = 4;
		placefixedM = placefixedM + 1;
end

if sd == 1
		KgrrM(placefreeM,1:DOF) = Kstarsixt(5,1:DOF);
		DispM(placefreeM,1) = UM_DISP(5,1);
		FrM(placefreeM,1) = 5;
		placefreeM = placefreeM + 1;
else
		KgrrM(placefixedM,1:DOF) = Kstarsixt(5,1:DOF);
		DispM(placefixedM,1) = UM_DISP(5,1);
		FrM(placefixedM,1) = 5;
		placefixedM = placefixedM + 1;
end

if md == 1
		KgrrM(placefreeM,1:DOF) = Kstarsixt(6,1:DOF);
		DispM(placefreeM,1) = UM_DISP(6,1);
		FrM(placefreeM,1) = 6;
		placefreeM = placefreeM + 1;
else
		KgrrM(placefixedM,1:DOF) = Kstarsixt(6,1:DOF);
		DispM(placefixedM,1) = UM_DISP(6,1);
		FrM(placefixedM,1) = 6;
		placefixedM = placefixedM + 1;
end
```

```
KgrM = zeros(6,6)
placefreeM = 1
placefixedM = num_freeM + 1

nn = ned_node
ad = node_data(9,nn)
sd = node_data(10,nn)
md = node_data(11,nn)

if ad == 1
        KgrM(:,placefreeM) = KgrrM(:,1);
        placefreeM = placefreeM + 1;
else
        KgrM(:,placefixedM) = KgrrM(:,1);
        placefixedM = placefixedM + 1;
end

if sd == 1
        KgrM(:,placefreeM) = KgrrM(:,2);
        placefreeM = placefreeM + 1;
else
        KgrM(:,placefixedM) = KgrrM(:,2);
        placefixedM = placefixedM + 1;
end

if md == 1
        KgrM(:,placefreeM) = KgrrM(:,3);
        placefreeM = placefreeM + 1;
else
        KgrM(:,placefixedM) = KgrrM(:,3);
        placefixedM = placefixedM + 1;
end
```

## Forces due to Temperature Change

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% f_TEMP.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


elongation = gradT * deltaT * l;

FM_TEMP(1,1) = FM_TEMP(1,1) - kstarsix(1,1)*elongation*cos(Ang);
FM_TEMP(2,1) = FM_TEMP(2,1) - kstarsix(1,1)*elongation*sin(Ang);
FM_TEMP(4,1) = FM_TEMP(4,1) + kstarsix(1,1)*elongation*cos(Ang);
FM_TEMP(5,1) = FM_TEMP(5,1) + kstarsix(1,1)*elongation*sin(Ang);
FM_TEMPT = Csix * FM_TEMP;

FG_TEMP(5*neg_node-4:5*neg_node-3) = FG_TEMP(5*neg_node-4:5*neg_node-3) +
FM_TEMP(1:2,1);
FG_TEMP(5*pos_node-4:5*pos_node-3) = FG_TEMP(5*pos_node-4:5*pos_node-3) +
FM_TEMP(4:5,1);
```

## Global Matrix Arrangement

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% f_KgArr.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


Kgrr = zeros(DOF,DOF)
placefree = 1;
placefixed = num_free + 1;
nn = 1;

while nn <= no_nodes
        ar = node_data(9,nn);       % axial restraint
        sr = node_data(10,nn);      % shear restraint
        mr = node_data(11,nn);      % moment restraint
        tr = node_data(12,nn);      % torsion restraint
        br = node_data(13,nn);      % bi-moment restraint

  if ar == 1
        Kgrr(placefree,1:DOF)      = Kg(5*nn-4,1:DOF);
        FGr(placefree,1)  = FG(5*nn-4,1);
        Ur(placefree,1)            = 5*nn-4;
        placefree = placefree + 1;
  else
        Kgrr(placefixed,1:DOF)     = Kg(5*nn-4,1:DOF);
        FGr(placefixed,1) = FG(5*nn-4,1);
        Ur(placefixed,1)   = 5*nn-4;
        placefixed = placefixed + 1;
  end

  if sr == 1
        Kgrr(placefree,1:DOF)      = Kg(5*nn-3,1:DOF);
        FGr(placefree,1)  = FG(5*nn-3,1);
        Ur(placefree,1)            = 5*nn-3;
        placefree = placefree + 1;
  else
        Kgrr(placefixed,1:DOF)     = Kg(5*nn-3,1:DOF);
        FGr(placefixed,1) = FG(5*nn-3,1);
        Ur(placefixed,1)   = 5*nn-3;
        placefixed = placefixed + 1;
  end

  if mr == 1
        Kgrr(placefree,1:DOF)      = Kg(5*nn-2,1:DOF);
        FGr(placefree,1)  = FG(5*nn-2,1);
        Ur(placefree,1)            = 5*nn-2;
        placefree = placefree + 1;
  else
        Kgrr(placefixed,1:DOF)     = Kg(5*nn-2,1:DOF);
        FGr(placefixed,1) = FG(5*nn-2,1);
        Ur(placefixed,1)   = 5*nn-2;
        placefixed = placefixed + 1;
  end
```

```
  if tr == 1
          Kgrr(placefree,1:DOF)      = Kg(5*nn-1,1:DOF);
          FGr(placefree,1)   = FG(5*nn-1,1);
          Ur(placefree,1)            = 5*nn-1;
          placefree = placefree + 1;
  else
          Kgrr(placefixed,1:DOF)     = Kg(5*nn-1,1:DOF);
          FGr(placefixed,1) = FG(5*nn-1,1);
          Ur(placefixed,1)   = 5*nn-1;
          placefixed = placefixed + 1;
  end

  if br == 1
          Kgrr(placefree,1:DOF)      = Kg(5*nn,1:DOF);
          FGr(placefree,1)   = FG(5*nn,1);
          Ur(placefree,1)            = 5*nn;
          placefree = placefree + 1;
  else
          Kgrr(placefixed,1:DOF)     = Kg(5*nn,1:DOF);
          FGr(placefixed,1) = FG(5*nn,1);
          Ur(placefixed,1)   = 5*nn;
          placefixed = placefixed + 1;
  end

  nn = nn + 1;
end

Kgr = zeros(DOF,DOF);
placefree = 1;
placefixed = num_free + 1;
nn = 1;

while nn <= no_nodes
          ar = node_data(9,nn);       % axial restraint
          sr = node_data(10,nn);      % shear restraint
          mr = node_data(11,nn);      % moment restraint
          tr = node_data(12,nn);      % torsion restraint
          br = node_data(13,nn);      % bi-moment restraint

  if ar == 1
          Kgr(1:DOF,placefree)       = Kgrr(1:DOF,5*nn-4);
          placefree = placefree + 1;
  else
          Kgr(1:DOF,placefixed)      = Kgrr(1:DOF,5*nn-4);
          placefixed = placefixed + 1;
  end

  if sr == 1
          Kgr(1:DOF,placefree)       = Kgrr(1:DOF,5*nn-3);
          placefree = placefree + 1;
  else
          Kgr(1:DOF,placefixed)      = Kgrr(1:DOF,5*nn-3);
          placefixed = placefixed + 1;
  end
```

```
    if mr == 1
        Kgr(1:DOF,placefree)      = Kgrr(1:DOF,5*nn-2);
        placefree = placefree + 1;
    else
        Kgr(1:DOF,placefixed)     = Kgrr(1:DOF,5*nn-2);
        placefixed = placefixed + 1;
    end

    if tr == 1
        Kgr(1:DOF,placefree)      = Kgrr(1:DOF,5*nn-1);
        placefree = placefree + 1;
    else
        Kgr(1:DOF,placefixed)     = Kgrr(1:DOF,5*nn-1);
        placefixed = placefixed + 1;
    end

    if br == 1
        Kgr(1:DOF,placefree)      = Kgrr(1:DOF,5*nn);
        placefree = placefree + 1;
    else
        Kgr(1:DOF,placefixed)     = Kgrr(1:DOF,5*nn);
        placefixed = placefixed + 1;
    end

    nn = nn + 1;
end
```

## *Acknowledgments*

The completion of this thesis would not have been possible had it not been for the help and support of a whole lot of people. Listed in order of appearance:

My parents (Ming and Gloria King) for raising me. Their support, guidance, and constant nagging, despite doubts and hurdles encountered in the author's earlier years, have actually paid off. Of course, I am a little biased. I owe them everything.

My brothers (the fellas - Warren and David) and sister-in-law (Rochelle) for spending more time worrying about my issues than I do. Advice and food have always been given freely between us, and I hope that never changes. We will all be taking our lives down separate paths and in doing so, the distance between us increases. Even though we're about to be separated by thousands of miles, I know that we'll always be close.

The gang (Patricia, Hector, Mia, and Mark), my family of classmates in the MS Structural Group (Carlos, Casey, Emma, Salvatore, Gilbert, David, Larry, Abel, and Andrew), and the friends I met over the past year (Ike, Helen, Adrian, and Adam) for providing the distractions and the laughter that made my stay at MIT one of the best years I've ever had. Thanks to you, I'll be leaving Boston with a lot of great memories that I'll never forget and a lot of great friends that I will never lose touch with.

My teacher, mentor and advisor (Prof. J.J. Connor), whose signature released me from academia and whose advice guided me into the working world. I should confess; I didn't always understand the material you presented in class, but our discussions outside of class have provided me with insights far more important than any academic specificities.

and of course,

The Editor Queen and my close friend (Jen Peltz) for helping me complete the most difficult part of this presentation. She took the time to read over this entire thesis and transformed an unintelligible collection of structural analysis terms into a semi-readable thesis. That which remains incomprehensible and grammatically incorrect are purely the fault of the author.

## *Bibliography*


Al-Mashary, F. and Chen, W.F. (1990a) Elastic second-order analysis for frame design, *Journal of Constructional Steel Rsearch,* **15**.

Atsuta, T. and Chen, W.F. *Theory of Beam - Columns, Vol. 1: In-Plane Behavior and Design*, New York, NY: McGraw-Hill, 1976.

Atsuta, T. and Chen, W.F. *Theory of Beam - Columns, Vol. 2: Space Behavior and Design*, New York, NY: McGraw-Hill, 1977.

Chen, W.F. and Lui, E.M. *Stability Design of Steel Frames*, Boca Raton, FL: CRC Press, 1991.

Chen, W.F. and Lui, E.M. *Structural Stability -Theory and Implementation*, New York, NY: Science Publishing Co., 1987.

Connor, J. J. and Klink B. *Introduction to Motion Based Design*, Southampton, UK: Computational Mechanics Publications, 1996.

Connor, J.J. and Pouangare C.C. (1995) A new method for the design of tall buildings: the MIT Design Method, *The Structural design of Tall Buildings,* **4**.

Felton, L.P. and Nelson, R.B. *Matrix Structural Analysis,* New York, NY: John Wiley & Sons, Inc., 1997.

Holzer, S.M. *Computer Analysis of Structures: Matrix Structural Analysis / Structural Programming,* New York, NY: Elsevier Science Publishing Co., 1985.

Stafford Smith, B., Kuster, M., and Hoenderkamp J.C.D. (1981) A generalized approach to the deflection analysis of braced frame, rigid frame, and coupled wall structures, *Canadian Journal of Civil Engineering,* **8**.

West, H.W. *Fundamentals of Structural Analysis,* New York, NY: John Wiley & Sons, Inc., 1993.

Wilson, E.L. and Habibullah, A. *SAP90 Structural Analysis User's Manual, Vol. 1*: 1992